turbolinux
Powerful Thinking.

turbolinux®

server 6 for zSeries and S/390

USER GUIDE

Turbolinux Server 6 for zSeries and S/390 User Guide
Version 6 December 2000

© 1999-2000 Turbolinux Inc. All Rights Reserved.

The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Turbolinux Inc. Turbolinux assumes no responsibility or liability for any errors or inaccuracies that may appear in this book.

This publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means -- electronic, mechanical, recording, or otherwise without the prior written permission of Turbolinux Inc., as long as this copyright notice remains intact and unchanged on all copies.

Turbolinux, Inc., Turbolinux, and Turbolinux logo are trademarks of Turbolinux Incorporated. All other names and trademarks are the property of their respective owners.

Written and designed at Turbolinux Inc.
8000 Marina Boulevard, Suite 300
Brisbane, CA 94005 USA
T. 650.228.5000
F. 650.228.5001
http://www.turbolinux.com/

TABLE OF CONTENTS

PREFACE vii

About This Guide viii
Contacting Us. x
Technical Support xi

CHAPTER 1 BASIC COMMANDS 1-1

Logging In. 1-2
Managing User Accounts 1-4
Closing a Linux Session 1-8
Working with Files and Directories 1-10
Managing System Processes 1-25
Managing Devices 1-28
Listing and Installing Packages 1-33
Accessing Online Manuals. 1-35

CHAPTER 2 APPLICATIONS 2-1

RPM Package Operations. 2-3
Domain Name Server (BIND) 2-6
Firewall Maintenance (ipchains) 2-25
File Transfer Protocol Client (TFTP). 2-29
File Transfer Protocol Server (ProFTP) 2-31
http Server (Apache) 2-35
Lightweight Directory Access Protocol (OpenLDAP) 2-44
Network File System (NFS) 2-49
Network Information Service (NIS). 2-59
OS Integration (Samba) 2-64
Printing Facilities (LPRng) 2-78
Programming Languages (perl, Python) 2-82

Proxy/Caching Server (Squid) 2-87
RPC Program Number Converter (Portmap) 2-91
SQL Database Management System (PostgreSQL) 2-94
SQL Database Server (MySQL) 2-101
Super Server (inetd) 2-107

Index I-1

PREFACE

Thank you for choosing Turbolinux Server 6 for zSeries and S/390 over the other versions of Linux currently available in the marketplace. Indeed it is an excellent choice. We at Turbolinux, Inc. have been and continue to work hard to make Turbolinux a high performance, comprehensive, easy to install, and yet easy to use product.

Turbolinux, which has been the Linux leader on the Pacific Rim, is now taking the world by storm. We have been working with Linux since 1993 and offered our own distribution in 1997 in both English and Japanese language versions. Now we offer Turbolinux in Brazilian Portuguese, Chinese, French, German, Italian, Japanese, Russian, Spanish, UK English, and US English languages.

For the latest information about Turbolinux and our fast-growing company, please visit our web site at http://www.turbolinux.com/.

Our success and your satisfaction with Turbolinux are all made possible through the magic of the Open Source movement and the original creator of Linux, Linus Torvalds. We want to thank Linus Torvalds and the countless developers around the world who have and continue to contribute to making the magic possible.

About This Guide

The *Turbolinux Server 6 for zSeries and S/390 User Guide* provides all the information you need to administer and use Turbolinux Server 6 for zSeries and S/390.

This guide is intended for use by Linux system administrators, users, and other personnel who need to understand how to use Turbolinux Server 6 for zSeries and S/390.

How to Use This Guide

Please use this book in conjunction with the IBM documents *Preparing for Installing Turbolinux for S/390* and *Installing Turbolinux for S/390* pdf files available at http://www.s390.ibm.com/linux/installfest/.

Guide Organization

This guide is organized into this preface and the following chapters:

- Chapter 1, *Basic Commands*, describes the basic commands of the Linux operating system including logging in; managing accounts, devices, and system processes; working with file directories, installing software packages, and accessing online manuals.
- Chapter 2, *Applications*, explains most of the main features of the Turbolinux server controls over services and daemons. It describes the configuration requirements for internetworking super daemon inetd, tcp_wrappers, name server, web server, samba, nfs, squid, MySQL, and OpenLDAP.

Typographical Conventions

This guide uses the following conventions:

- *Italics* indicate CD and book titles, and emphasize words.
- `Monospace` indicates utilities, man help, commands, programs, and text examples that need to be entered exactly as shown.
- File names, directory paths, and daemons are shown in Arial font.
- Buttons, menu items, or options are indicated by initial capital letters.
- Other options, messages, and default settings are indicated by double quotes. For example: If you select Save Profile, the message "Profile save failed" will appear.
- The system prompt # distinguishes the root user from an ordinary user, who is designated by a $.

NOTE Take care not to mistake the # representing the prompt for the root user from a # which occasionally indicates a remark line in script. The two are identical.

Contacting Us

We at Turbolinux, Inc. are constantly trying to improve our products and documentation. We invite and value your feedback regarding any documentation inaccuracies, insufficiencies, errors, or omissions. We will fix such problems in the next release.

Please email us your feedback on any aspect of our products at s390@turbolinux.com.

Technical Support

For more information about support, services, and products from Turbolinux, please go to http://www.turbolinux.com/s390/.

Email sales@turbolinux.com for more information about support packages available for purchase. Please specify that you would like to purchase support for Turbolinux Server 6 for zSeries and S/390.

If you have purchased a support package and require support, contact your usual technical support representative.

We hope you enjoy running Turbolinux on your mainframe and look forward to providing you with the high quality service that you expect from your mainframe vendor.

CHAPTER 1

BASIC COMMANDS

This chapter introduces some of the basic commands of the Linux operating system. It explains the process and commands associated with:

- *Logging In*, page 1–2
- *Managing User Accounts*, page 1–4
- *Closing a Linux Session*, page 1–8
- *Working with Files and Directories*, page 1–10
- *Managing System Processes*, page 1–25
- *Managing Devices*, page 1–28
- *Listing and Installing Packages*, page 1–33
- *Accessing Online Manuals*, page 1–35

Logging In

Like other forms of Unix, you must log in to Linux before you can use it. Because Linux is a multiuser system, each user must be uniquely identified. The login procedure establishes the identity of each user. This allows a legitimate user access to the system and guards against entry by an unauthorized user.

During the Turbolinux installation, the root account is created. One user account can be created during installation. Additional user accounts can be created after installation is complete.

The "#" at the end of the prompt line distinguishes the root user from an ordinary user, who is designated by a "$."

For more information on the differences between a root user and normal user, see *Managing User Accounts* on page 1-4.

Logging In

Almost all access to the Turbolinux server is done remotely through `telnet` or `ssh`. Occasionally, the word "terminal" is used in this document to describe the interface on the remote machine through which the user connects to the server.

At the terminal, the login prompt displays:

```
[hostname] login:
```

1. Enter a user name here and press <ENTER>.

   ```
   Password:
   ```

2. The system asks for a user password. Enter the password that has been assigned to the user. (For security reasons, the system does not display the password string.)

   ```
   [username]@[hostname] [currentdirectory]$
   ```

A proper login prompt line is shown above. For example, if the hostname is [Turbo], and the login user is [root], the line appears as follows:

```
[root@Turbo /root]#
```

Logging Out

Before you can close your Turbolinux session, you need to log out.

Use the logout command from the command line.

1. To log out, enter the command:
   ```
   [root@Turbo /root]# logout
   ```
2. Press <ENTER>.

Shutting Down

To conclude your Turbolinux session without logging out first, use the shutdown command as described in *Closing a Linux Session* on page 1-8.

Managing User Accounts

In Linux there are several references to *root*. These references are:

- The root account
- The root account's login directory
- The root directory, which is represented by a single forward slash (/).

The "root" account has superuser privileges. Root has the greatest power and authority within the system. When you log in as the root user you can access, configure, and run anything. While this gives you a great amount of power, it can also be risky to always log in as the root user. You may not be able to recover from configuring something incorrectly or deleting an essential or important file.

To operate safely, log in as an ordinary user and switch (see *Switching to Another User (su)* on page 1-7) to a superuser when necessary. When you have completed the task you need to perform as a superuser with root privileges, log out of the root account.

NOTE The root account and password should be under the strict control of the person in charge of the system, who is usually a system administrator.

Creating and Changing User Accounts (useradd, passwd)

During the installation of Turbolinux Server 6 for zSeries and S/390, the root account is created. One user account can be created during installation. All additional user accounts can be created after installation is complete.

NOTE Only the superuser can use the `useradd` and `passwd` commands.

You can create new user accounts with the commands `useradd` and `passwd`.

```
# useradd [User name to be added.]
```

1. Add a user account with the following command, using 'jon' as the user name:

   ```
   # useradd jon
   ```

   At this point, the password for the user account 'jon' has not been set so it would still be impossible for this user to actually log in.

Set the password for the user account just created as follows.

```
# passwd jon
```

The system displays:

```
Changing password for user jon
New UNIX password:
```

2. When the above message appears, set the password. The system displays:

   ```
   Retype new UNIX password:
   ```

3. Confirm identity by entering the password again.

   ```
   passwd: all authentication tokens updated successfully
   ```

   The above message indicates that the password has been set.

Changing a Password

To change an already existing password, type the `passwd` command and follow the prompt. For example to change user charles' password:

```
# passwd charles
```

The system displays:

```
Changing password for user charles
New UNIX password:
```

Handling Error Messages

When setting a password, you may receive one of the following messages:

BAD PASSWORD	`it is too short` You tried to enter a password string that was less than six characters in length.
BAD PASSWORD	`it is based on a dictionary word` You tried to enter a common name or word found in dictionaries.
BAD PASSWORD	`it is too simplistic/systematic` You tried to enter a password string that was too simplistic or systematic.

NOTE The `BAD PASSWORD` error message does not prevent the user from choosing the password. Retyping the password at the `Retype new UNIX password` prompt will set the password, bad or not.

Deleting User Accounts

You can delete a user account with the command `userdel`.

```
# userdel [User name to be deleted.]
```

Delete a user account with the following command, using the previously created user 'jon':

```
# userdel jon
```

Use the `-r` option to also delete the user's home directory.

NOTE Only the superuser is allowed to run the `userdel` command.

Switching to Another User (su)

You can switch from the current user to another user without doing a logout and a login.

```
$ su [The username you want to become.]
```

1. To switch from the currently logged user, 'jon' to a different user, 'charles', you would do it as follows.

```
$ su charles
Password
```

2. Enter the correct password for 'charles'.

If you don't want to continue to use the user environment of the current user, add a "-" (minus sign) followed by a space.

```
$ su - charles
```

Without a username, you will switch to 'root'.

```
$ su -
Password:
#
```

As root, you may switch to an ordinary user without entering a password.

You can find out what user you are with the following command.

```
$ whoami
jon
```

In the above example, the user is logged in as 'jon'.

You can return to the previous user with the following command:

```
[root@Turbo /root]# exit
```

Closing a Linux Session

You must log out before concluding a Turbolinux session.

Command Line Shutdown

1. Use the `logout` command from the command line.
   See *Logging Out* on page 1-3.
2. Use the `shutdown` command as described in this section.
3. Shutdown command will end the Linux session.

NOTE Only the superuser is allowed to run the `shutdown` command.

You can shut down your Turbolinux session with the following command.

```
# shutdown [options][time][message]
```

The following are examples of available options:

-h	stop (halt)
-r	reboot

Timed Shutdown

You can set up a timed shutdown ('time') as follows.

`xx.yy`	Execute shutdown at xx hours yy minutes.
`+x`	Execute shutdown after x number of minutes.
`now`	Execute shutdown immediately.

The 'message' may be omitted, or you can send a message to each logged in user. For example, a message such as, "The system will shutdown at 6 PM; please finish your work before then," is quite useful.

Closing Down a Turbolinux Session

You can close the Turbolinux session immediately with one of the following commands.

```
# shutdown -h now
```

Or,

```
# halt
```

With the `halt` command, all running processes are stopped and the following message appears on the screen:

```
The System is halted.
```

Restarting the System after Powering Down

If you want to immediately restart Turbolinux, use one of the following commands.

```
# shutdown -r now
```

Or,

```
# reboot
```

The computer displays a message stating that all running processes have been stopped and restarts automatically afterward.

Working with Files and Directories

Linux keeps track of files and folders by requiring unique names and by using a tree-like structure to store them. This section explains how you can get a listing of the files and folders or directories on your system, move to another directory, copy or move files or directories, create new directories, and remove directories.

Listing Files and Directories (ls)

The `ls` (list files) command is used to get information on files and directories.

```
$ ls [options] [file name] [directory name]
```

Some typical options are:

-l	Lists not only file names alone, but also full details about each file.
-a	All files are listed, including normally hidden files.
-F	A symbol is affixed to file names to show the file type. (Executable files are marked with an asterisk "*", directories are marked with an backslash "/".)

NOTE In Turbolinux, `ls` is set as an alias for `ls -F`.

If no file or directory names are given, the files and directories of the current directory are listed.

For example, see below how the various options work with /home/jon.

No options — In Turbolinux, this is equivalent to the -F option.

```
$ ls /home/jon
core nsmail/
```

The files and directories names alone are shown.

-l Option

```
$ ls -l /home/jon
    total 352
    -rw------- 1 jon jon 356352 Aug 27 07:25 core
    drwx------ 2 jonjon1024 Aug 27 01:01 nsmail/
```

Detailed information for each file and directory is also shown.

-a Option

The table below shows the results of the ls command with the -a option. All files and directories in the current directory, including hidden files and directories and subdirectories are shown.

```
$ ls -a /home/jon
```

./	bashrc	.lang	.xsession
../	.elvisrc	.less	1
..Xdefaults	.emacs	.rhosts	2
bash_history	.emacs.el	.screenrc	jon1/
bash_logout	.exrc	.tcshrc	
.bash_profile	.inputrc	.vimrc	

Changing to a Different Directory (cd)

Use the cd (Change working Directory) command to change the current directory to another one.

```
# cd [name of the desired directory]
```

With no directory name, cd changes the directory to the user's home directory.

It is not necessary to specify the full path to the desired directory name. The following arguments can be used.

.	current directory
..	the directory above the current directory
~	user's home directory
-	the directory immediately previous to the current directory

This switches from the current directory (/home/jon) to the /home directory.

```
$ cd /home
$ cd ..
```

The two commands above accomplish the same thing.

Next, switch from the current directory (/home) to the user's home directory which was the directory immediately previous to the current directory.

```
$ cd /home/jon
$ cd ./jon
$ cd jon
$ cd ~
$ cd -
$ cd
```

All of the six commands above accomplish the same thing.

Use the pwd (Print Working Directory) command to find out what your current directory is.

```
$ pwd
/home/jon
```

Copying Files (cp)

Use the cp (CoPy files) command to copy files and directories.

```
$ cp [options] [source filename | source directory name]
    [destination filename | destination directory name]
```

Some typical options are:

-b	If the destination file already exists, that file is backed up before the copy proceeds.
-f	If the destination file already exists, that file is forceably overwritten.
-i	If the destination file name already exists, you are asked if you want to overwrite that file or not. If the answer is 'y', the file is overwritten. For any answer besides 'y', the copy does not proceed.

NOTE In Turbolinux, cp is set as an alias for cp -i.)

-u	If the destination file already exists, the copy proceeds only when the date on the destination file is older than the source file. (If the destination file is newer, copy does not proceed.)
-p	Preserve the attributes of the source file (date, ownership, permissions) in the copy.
-v	The results of the copy (source filename to destination filename) are shown.
-R	Copy directories recursively.

For example, copy file1.txt to file2.txt.

```
$ cp -v file1.txt file2.txt
file1.txt -> file2.txt
```

With the -v option, the results of the copy are shown.

Copy file1.txt into the ../public directory.

```
$ cp -iv file1.txt ../public
cp: overwrite '../public/file1.txt'? y
file1.txt -> ../public/file1.txt
```

Because a file with the same filename already exists, you were asked for permission to overwrite it and answered 'y', so the copy proceeded, and the results of the copy are shown.

For directories, you can copy directory1 to directory2.

```
$ cp -R directory1/ directory2/
$ ls
directory1/directory2/
```

The entire directory was copied.

Moving Files (mv)

Use the mv (MoVe files) command to move files and directories.

```
$ mv [options] [source filename | source directory name]
    [destination filename | destination directory name]
```

Some typical options are:

-b	If the destination file already exists, that file is backed up before the move proceeds.
-f	If the destination file already exists, that file is forceably overwritten.
-i	If the destination file already exists, you are asked if you want to overwrite that file or not. If the answer is 'y', the file is overwritten. For any

answer other than 'y', the move does not proceed.

NOTE In Turbolinux, `mv` is set as an alias for `mv -i`.

-u — If the destination file already exists, the move proceeds only when the date on the destination file is older than the source file. (If the destination file is newer, move does not proceed.)

-v — The results of the move (source filename to destination filename) are shown.

Move file1.txt into the ../public directory.

```
$ mv -iv file1.txt ../public
mv: overwrite '../public/file1.txt'? y
file1.txt -> ../public/file1.txt
```

Because a file with the same filename already exists, you were asked for permission to overwrite it and answered 'y', so the move proceeded, and the results of the move are shown. If you are moving directories, the move cannot proceed if a destination directory with the same name already exists.

Changing File Names (mv)

You also use the same mv (MoVe files) command to change the name of files or directories.

```
$ mv [options] [source filename | source directory name]
    [destination filename | destination directory name]
```

Here is a typical option:

-v — The results of the name change (source filename to destination filename) are shown.

For example, change the name of file1.txt to file2.txt.

```
$ mv -v file1.txt file2.txt
file1.txt -> file2.txt
```

With the `-v` option, the results of the name change are shown.

Creating a Directory (mkdir)

Use the `mkdir` (MaKe DIRectory) command to create new directories.

```
$ mkdir [options] [name of the new directory]
```

Here is a typical option:

-m	Set permissions at the same time you create the new directory.

For example, create a new directory (directory2) under the current directory.

```
$ mkdir directory2
$ ls
directory2/
```

New directory2 was created.

Deleting Files and Directories (rm, rmdir)

Use the `rm` (ReMove files) command to delete files and directories.

```
$ rm [options] [name of file to delete | name of directory to
    delete]
```

Typical options:

-f	Forces deletion of a file or files for which the user does not have write permission.
-i	For each file, you are asked if you want to delete that file or not. If the answer is 'y', the

	file is deleted. For any answer other than 'y', the delete does not proceed.
-V	The results of the delete are displayed.
-r	Recursively deletes all files, subdirectories and directories.

For example, delete file1.txt that resides under the current directory.

```
$ rm -i file1.txt
rm: remove 'file1.txt'? y
```

Because you answer 'y' to the question whether to remove the file, it was deleted.

Delete the directory /home/directory1.

```
$ rm -riv /home/directory1/
rm: descend into directory '/home/directory1'? y
    removing all entries of directory /home/directory1
    rm: remove '/home/directory1/file1.txt'? y
    removing /home/directory1/file1.txt
    rm: remove directory '/home/directory1'? y
    removing the directory itself: /home/directory1
```

Because you answered 'y' to the question of descending into the directory and removing its contents, each directory was removed and the results of the delete are displayed.

Use the `rmdir` (ReMove DIRectory) command to delete empty directories.

```
$ rmdir directory2
```

In this case, you are not asked for confirmation before the removal proceeds.

Viewing Text Files (cat, less, more)

Use the `cat` (conCATenate) command to view the contents of text files.

```
$ cat [options] [name of file to view]
```

A typical option is:

-n Display line numbers.

For example, display the contents of /etc/lftp.conf. The -n option adds the line numbers to the display. (Note: Lines remarked out with a # have been omitted.)

```
$ cat -n /etc/syslog.conf

     4 /dev/console
     7 *.info;mail.none;new.none;authpriv.none/var/log/messages
    10 authpriv.#      /var/log/secure
    13 mail.*          /var/log/maillog
    17 *.emerg         *
    21 uucp,news.crit  /var/log/spooler
    24 local7.*        /var/log/boot.log
    29 news.=cret      /var/log/news/news.cret
    30 news.=err       /var/log/news/news.err
    31 news.notice     /var/log/news/news.notice
```

With cat, a long file will scroll off the screen and you will only be able to see the end of it. This is inconvenient. To page through a file, use the less command.

```
$ less [options] [name of file to view]
```

When using less to view a file, there are a number of commands available from the keyboard. The main ones are:

<SPACE>	Scroll ahead one screen.
<ENTER>	Scroll ahead one line.
[q]	Interrupt display, exit.
[/][searchpattern]	Search backward from the current screen for 'search pattern'.
[n]	Repeat search.
[d]	Scroll a half screen ahead.

[h]	Display help.
[w]	Scroll backward one screen.
[u]	Scroll backward half screen.
[y]	Scroll backward one line.
[?] [string pattern]	Search backward for [string pattern] starting from the current screen.
[N]	Repeat the previous search backward, starting from the current screen.

For example, display the contents of the file /etc/logrotate.conf. Go to that directory and type:

```
$ less /etc/logrotate.conf

    # see “man logrotate” for details
    # rotate log files weekly
    weekly
    # keep 4 weeks worth of backlogs
    rotate 4
    # send errors to root
    errors root
    # create new (empty) log files after rotating old ones
    #create
    # truncate the original log file in place after creating a
    copy
    copytruncate
    # compress log file copies
    compress
    #don’t complain about missing log files
    missingok
    # RPM packages drop log rotation information into this
    directory
    include/etc/logrotate.d
    --Less--(28%)
```

If only a part of the file fits onto one screen, a line appears at the bottom saying, "-- Less -- (xx%)" in which "xx" is the percentage of the file that has been displayed.

An abbreviated version of the less command is called `more`.

```
$ more [options] [name of file to view]
```

Finding Files (find)

Use the `find` command to search for files.

```
$ find [options] [path to search target] [expressions]
```

Typical 'find' expressions are:

-name [string pattern]	Search for files that match [string pattern].
-iname [string pattern]	Search for files that match [string pattern] ignoring the distinction between upper and lower case.
-path [string pattern]	Search for files that match [string pattern] which includes the full path name.
-ipath [string pattern]	Search for files that match [string pattern] which includes the full path name and ignores the distinction between upper and lower case.
-uid [user ID]	Set the file's numeric user ID and search.
-user [user name]	Set the file's owner [user name] and search.
-gid [group ID]	Set the file's numeric group ID and search.

-group [group name]	Set the file's group [group name] and search.

For example, search for all files that end with .conf under /etc.

```
$ find /etc -name "*.conf"
    /etc/host.conf
    /etc/inetd.conf
    /etc/nsswitch.conf
    /etc/ld.so.conf
          <lines omitted>

    /etc/named.conf
    /etc/irquery.conf
    /etc/resolv.conf
    /etc/modules.conf
```

You can use wildcards "*" and "?", etc., when setting the string pattern.

Searching for Strings (grep)

Use the grep (Global Regular Expression Printer) command to search for a text string within text files.

```
$ grep [options] [string pattern for search] [target files]
```

Typical options are:

-i	Ignore the distinction between upper and lower case in the search.
-l	From the usual search results, display only the name of the file.
-n	Display line numbers.
-x	Search for only those matches that match the whole string pattern line.

For example, to search for the string pattern ‘prompt’ in all the files under /etc/lftp.conf:

```
$ grep -n prompt /etc/lftp.conf

    7:## make prompt look better
    8:set prompt “lftp \u\@\h:\w “
```

The `-n` option adds the line numbers to the display of the search results.

Compressing and Expanding Files (gzip)

Use `gzip` (GNU Zip) to compress or expand gzip (file extension .gz) type files.

```
$ gzip [options] [file name]
```

Typical options are:

-d	Decompress the file.
-f	Force an overwrite of a file with the same name.
-v	Verbosely display the results.

For example, compress all the `.txt` files under the current directory.

```
$ gzip -v *.txt

    file1.txt:         -82.6% -- replaced with file1.txt.gz
    file2.txt:         -53.0% -- replaced with file2.txt.gz
    file3.txt:         -72.2% -- replaced with file3.txt.gz
    file4.txt:         -75.3% -- replaced with file4.txt.gz
    file5.txt:         -66.5% -- replaced with file5.txt.gz
```

When the ".gz" extension has been affixed to each file, that file has been compressed. With the `-v` option, the effect of the operation is also displayed.

Next, expand all the files that were compressed with the previous command.

```
$ gzip -dv *.gz

    file1.txt.gz:       -82.6% -- replaced with file1.txt.gz
    file2.txt.gz:       -53.0% -- replaced with file2.txt.gz
    file3.txt.gz:       -72.2% -- replaced with file3.txt.gz
    file4.txt.gz:       -75.3% -- replaced with file4.txt.gz
    file5.txt.gz:       -66.5% -- replaced with file5.txt.gz
```

Creating and Extracting Archives (tar)

Use the `tar` (Tape file ARchiver) command to create an archive or to extract a file or files from an archive.

```
$ tar [options] [file name of archive] [target file name]
```

Typical options are:

-c	Create a new archive.
-f	Use file name to create an archive.
-v	Verbosely list files processed.
-x	Extract files from an archive.
-z	Compress or extract files through gzip.

For example, create a new archive of all the txt files under the current directory.

```
$ tar -cvf file.tar *.txt

    file1.txt
    file2.txt
    file3.txt
    file4.txt
    file5.txt
```

Next, extract all the files from the archive just created.

```
$ tar -xvf file.tar

    file1.txt
    file2.txt
    file3.txt
    file4.txt
    file5.txt
```

Files are often seen with the .tar.gz extension. Such files have been compressed with `gzip` after they were archived. They can be decompressed and extracted as follows.

```
$ tar -zxvf file.tar.gz

    file1.txt
    file2.txt
    file3.txt
    file4.txt
    file5.txt
```

Managing System Processes

User initiated processes, server services running in the background, and daemons are all included under the broad meaning of the term “process.” Because Linux is a multi-tasking, multi-user operating system, process management is necessary. Process management commands are explained below.

Finding Out the State of Processes (ps)

Use `ps` (Process Status) to find out what processes are running.

```
$ ps [options]
```

Typical options are:

-a	Display all processes initiated by the current terminal.
-f	Display the process hierarchy in a tree format.
-l	Display a detailed listing (long format).
-u	Display the information in a user-preferred format
-x	Display also processes not initiated on the current terminal (tty).

For example, you can display all the processes that are currently running. (Note: This example shows only a partial listing of all running processes.)

```
$ ps -ax
```

PID	TTY	STAT	TIME	COMMAND
1	?	S	0:020	init [3]
2	?	SW	0:00	[kmcheck]
3	?	SW	0:00	[kflushd]
4	?	SW	0:04	[kupdate]
5	?	SW	0:00	[kpiod]
6	?	SW	0:00	[kswapd]
159	?	S	0:06	syslogd -m 0
167	?	S	0:00	klogd
177	?	S	0:00	/usr/sbin/atd
187	?	S	0:01	crond
197	?	S	0:00	inetd
201	?	S	0:00	/usr/sbin/sshd

To display all the current processes in tree format use the command:

```
$ ps -axf
```

Stopping Processes (kill)

Use the `kill` (KILL process) command to terminate processes.

```
$ kill [options] [process ID]
```

The ‘process ID’ is a unique number attached to each process. The “PID” shown by the `ps` command refers to this number.

Typical options are:

-l	List all signal names.
-[signal number]	Send the signal specified by the signal number to the process.

Use the `-l` option To display a list of all available signal numbers and names. (Note: The result displayed below is a partial list.)

```
$ kill -l
```

```
1)  SIGHUP     2)  SIGINT   3)  SIGQUIT   4)  SIGILL
5)  SIGTRAP    6)  SIGABRT  7)  SIGBUS    8)  SIGFPE
9)  SIGKILL    10) SIGUSR1  11) SIGSEGV   12) SIGUSR2
13) SIGPIPE    14) SIGALRM  15) SIGTERM   17) SIGCHLD
18) SIGCONT    19) SIGSTOP  20) SIGTSTP   21) SIGTTIN
22) SIGTTOU    23) SIGURG   24) SIGXCPU   25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF  28) SIGWINCH  29) SIGIO
30) SIGPWR     31) SIGSYS   32) SIGRTMIN  33) SIGRTMIN+1
```

If the signal number is omitted, the kill command sends software termination signals (signal number "15) SIGTERM").

For example, terminate the process with process ID 555.

```
$ kill 555
```

If you want to force a process to terminate, send signal number "9) SIGKILL".

```
$ kill -9 555
```

To execute the `kill` command on a process, you must be the owner of that process or superuser.

Managing Devices

As Turbolinux is a flavor of Unix, it treats the mouse, hard drives, peripheral devices, etc., as files.

Turbolinux, like Unix, is organized in a tree structure starting with root (/) as the base. As stated, devices are treated as files. The devices exist in the form of an interconnected tree.

Partitions

A single physical disk must be divided into one or more *partitions*, which are logically distinct areas on a physical disk.

A partition is a logical allocation of a portion of a single physical disk. By partitioning, the system can treat one disk as if it were multiple disks. For more information, see Setting up the Target Linux System on page 37 of the *Preparing for Installing Turbolinux for S/390* file found at http://www.s390.ibm.com/linux/installfest/.

Since there is no concept of a drive in Turbolinux, devices are treated as files, as shown below.

DASD hard drives:

/dev/dasda

/dev/dasdb

/dev/dasdc

... ordered by SCSI ID

Root and Swap Partitions

It is common practice to create several partitions for a single system when you install Turbolinux. At the very least, you will need to create a swap partition. The purpose of the swap partition is to allow

for the use of 'virtual memory'. The swap partition can not be accessed by the user.

In many cases it is useful to prepare other partitions in addition to the root partition and swap partition. How these other partitions are set up will vary depending on what type of Linux system you run.

For example, if there will be many login users, a great deal of space must be made available under /home. In this case you may want to create a separate partition that will allow a more efficient use of the large capacity hard disk required. In order to make it easy to backup your data, you might want to consider creating a partition for backups only, thus separating it from the root directory.

There is no single pattern that works unconditionally. Partition your disks to fit the system you want to create.

Finding Out Disk Usage (df, du)

Use the `df` (Disk Free) command to find out how much space has been used on each partition.

```
$ df [options] [target device name | target partition name |
    target directory name | target file name]
```

This command displays the total capacity, amount used, amount available, percentage used in block (1024bytes) units as well as the mount point. When you omit the targets, the currently mounted partitions are displayed. Typical options are:

-a	Display information on all file systems.
-k	Display sizes in kilobytes.
-m	Display sizes in megabytes.
-h	Display sizes with a G (for gigabytes) or M (for megabytes) suffix.

-H The same as the -h option except that sizes are displayed in 1000 byte units instead of 1024 byte units.

-i Display sizes in inodes.

Compare the results of using various options in the following listings.

```
$ df
Filesystem   1k-blocks   Used     Available  Use%   Mounted on
/dev/dasda1  708652      48456    624200     7%     /
/dev/dasdc1  6382148     874988   5182960    14%    /usr

$ df -a
Filesystem   1k-blocks   Used     Available  Use%   Mounted on
proc         0           0        0          -      /proc
/dev/dasda1  708652      48456    624200     7%     /
/dev/dasdc1  6382148     874988   5182960    14%    /usr

$ df -k
Filesystem   1k-blocks   Used     Available  Use%   Mounted on
/dev/dasda1  708652      48456    624200     7%     /
/dev/dasdc1  6382148     874988   5182960    14%    /usr
```

```
$ df -m
Filesystem   1M-blocks   Used     Available  Use%   Mounted on
/dev/dasda1  692         47       610        7%     /
/dev/dasdc1  6233        854      5061       14%    /usr
```

The du (Disk Usage) Command

Use the du (Disk Usage) command to find out how much space is occupied by each directory.

```
$ du [options] [target directory name | target file name]
```

Under the specified directory, the size of each file is shown in blocks. If the target is omitted, information on the current directory is displayed.

Typical options are:

-a	Display counts for all files, not only directories.
-b	Display size in bytes.
-c	Display the total size of the target on the last line.
-k	Display size in kilobytes.
-h	Display sizes with a G (for gigabytes) or M (for megabytes) suffix.
-H	The same as the -h option except that sizes are displayed in 1000 byte units instead of 1024 byte units.

Compare the results of using various options in the following listings.

```
$ du /usr/lib/bind
44          ./include/arpa
56          ./include/isc
4           ./include/net
4           ./include/sys
168         ./include
1268        ./lib
1440        ./
```

```
$ du -b /usr/lib/bind
45056       ./include/arpa
57344       ./include/isc
4096        ./include/net
4096        ./include/sys
172032      ./include
1298432     ./lib
1474560     ./
```

Listing and Installing Packages

Turbolinux uses RPM (Red Hat Package Manager) as its package manager. RPM provides an efficient management environment in which to install, uninstall, update, or examine packages. It also manages the interdependencies that exist between packages.

Packages are stored on the *Turbolinux Server 6 for zSeries and S/390 Install CD* in /RPMS/s390. Source packages are in /SRPMS.

NOTE Some packages require a superuser to install them.

The syntax of the RPM command is as follows:

```
$ rpm [options] [RPM package name]
```

Typical options are:

-i	Install.
-U	Upgrade.
-e	Erase.
-h	Display progress status with hash marks.
-v	Verbose display. (Use with -h for a nice display.)
-q	Query to see what packages are currently installed.

Various scenarios using `apache 1.3.14-1.src.rpm` are shown below.

To get information on the currently installed RPM package:

```
# rpm -q apache
apache-1.3.14-1
```

If the package is not installed, the following message appears.

```
# rpm -q apache
package apache-1.3.14-1 is not installed
```

Use this command to see a list of all the files associated with a currently installed RPM package.

```
# rpm -ql apache

    /etc/httpd/conf/access.conf
    /etc/httpd/conf/httpd.conf
    /etc/httpd/conf/magic

          <lines omitted>

    /usr/share/man/man1/dbmmanage.1.gz
    /usr/share/man/man1/htdigest.1.gz
    /usr/share/man/man1/htpasswd.1.gz
    /usr/share/man/man8/httpd.i.gz
```

To uninstall, use this command:

```
# rpm -e apache
```

Use this command to see detailed information on an RPM file.

```
# rpm -qi apache
```

Name	:	apache	Relocations	:	(not relocateable)
Version	:	1.3.14	Vendor	:	(none)
Release	:	1	Build Date	:	Sun Nov 12 14:55:29 2000
Install date	:	Wed Dec 6 12:02:32 2000	Build Host	:	s390.dev.us.tlan
Group	:	Networking/Daemons	Source RPM	:	apache-1.3.14-1.src.rpm
Size	:	4904533	License	:	Freely distributable & usable
Summary	:	World Wide Web server (httpd)			
Description	:	Apache is a full-featured web server that is freely available, and also happens to be the most widely used on the Internet. Install this if you would like to run a web server.			

Accessing Online Manuals

In Linux, use the man command to obtain information about a program. Use it as follows:

```
$ man [options] [chapter number] [the name of the command you want
    information about]
```

Information can be viewed on the screen with the man command in the same way as with less.

For example, to view the manual page for the ls command:

```
$ man ls
```

Typical options are:

-a	List all the manual pages relating to the command you are looking for in sequence.
-w	List the full path of the locations of the manual pages that would be displayed.

For example, to display all the manual pages that are related to users, you would enter the following command:

```
# man -aw users
```

The return of this command is:

```
/usr/man/man1/users.1
```

Note that man is organized into the following sections.

1. Shell commands
2. System calls
3. Library calls
4. Special files
5. File formats and conventions
6. Games
7. Macro packages and conventions

8. System administration commands

9. Kernel routines

For example, `users` is a Shell command and is therefore explained in section 1.

To find information on the command `users`, you would do the following:

```
$ man users
```

For a detailed explanation on the use of `man`, please look at the man page on man itself.

```
$ man man
```

CHAPTER 2

APPLICATIONS

This chapter contains information on the applications bundled with Turbolinux Server 6 for zSeries and S/390.

The initial section, *RPM Package Operations* on page 2–3, is an introduction to various aspects of package management including installing, uninstalling, and package verification.

The sections after *RPM Package Operations* are organized alphabetically by the service they provide.

- *Domain Name Server (BIND)*, page 2–6

  Translates IP addresses into host names, and vice versa
- *Firewall Maintenance (ipchains)*, page 2–25

  Sets up, maintains, and inspects the firewall rules in the Linux kernel
- *File Transfer Protocol Client (TFTP)*, page 2–29

  Allows users to transfer files to and from a remote machine
- *File Transfer Protocol Server (ProFTP)*, page 2–31

  An ftp server which allows the transfer files between a PC and a remote computer using the ARPANET standard File Transfer Protocol
- *http Server (Apache)*, page 2–35

  The most widely used web server in the world

- *Lightweight Directory Access Protocol (OpenLDAP)*, page 2–44

  A directory service protocol designed to handle account information services
- *Network File System (NFS)*, page 2–49

  Makes it possible to share files between hosts that are connected to a network
- *Network Information Service (NIS)*, page 2–59

  Used to share information about computers on the network, such as login names, passwords, home directories and groups
- *OS Integration (Samba)*, page 2–64

  Integrates the Windows operating system with UNIX
- *Printing Facilities (LPRng)*, page 2–78

  Print filters, facilities, and tools
- *Programming Languages (perl, Python)*, page 2–82

  Two programming languages included with Turbolinux Server 6 for zSeries and S/390
- *Proxy/Caching Server (Squid)*, page 2–87

  Keeps metadata and especially hot objects cached in RAM, caches DNS lookups, supports non-blocking DNS lookups, and implements negative caching of failed requests
- *RPC Program Number Converter (Portmap)*, page 2–91

  Converts RPC program numbers to DARPA protocol numbers
- *SQL Database Management System (PostgreSQL)*, page 2–94

  An object-relational DBMS that supports almost all SQL constructs
- *SQL Database Server (MySQL)*, page 2–101

  a multi-user, multi-threaded SQL database server
- *Super Server (inetd)*, page 2–107

  A high-level program that acts as a server to other servers

RPM Package Operations

Turbolinux Server 6 for zSeries and S/390 comes on a CD-ROM with install, uninstall, and other programs. The programs run as shown below.

Install

1. Confirm that you are logged in as the root user.

```
# whoami
root
```

2. Insert the CD-ROM in the drive and mount it.

```
# mount /mnt/cdrom
```

3. Move to the directory containing the RPM package.

```
# cd /mnt/cdrom/RPMS/
```

4. Confirm the package name (represented by <xxxx>) and version.

```
# ls <xxxx>*
xxxx-xxx-xxxx-TLS.i386.rpm
```

5. Install the package.

```
# rpm -ivh xxxx-xxx-xxxx-TLS.i386.rpm
xxxx
```

6. Confirm the installation.

```
# rpm -qi xxxx
```

Uninstall

1. Use the -e option with the RPM command.

```
# rpm -e xxxx
```

2. To force an unconditional uninstall append the -nodeps switch.

```
# rpm -e xxxx --nodeps
```

Verify Packages

1. Use the -V option with the RPM command.

```
# rpm -V xxxxx
```

2. To verify all packages you can use:

```
# rpm -Va
```

This may produce rather verbose output so it may be best to redirect output to a file for later viewing. One way to to this is to follow this example:

```
# rpm -Va > /tmp/chk-allpkgs
```

What Packages are Installed?

Use the -qa option to query all installed packages. The following command will list all installed packages in alphabetical order:

```
# rpm -qa | sort | more
```

Daemons

In Unix terminology, *daemon* is customarily used to denote server programs. Daemons are memory-resident programs executed only when they receive a request from another program. Server programs such as FTP and TELNET are generally implemented as daemons. The names of most daemons end in “d” in the form xxxd.

Note: In this guide protocol names are shown in all upper-case letters, for example, TELNET and FTP. Client programs corresponding to these protocols are shown in lower-case letter, for example, telnet and ftp. While Internet protocols are precisely defined, the particular method of implementing them often is not precisely defined. Thus for the FTP protocol there are various program implementations such as `ftp` and `ncftp`, each with minor variations. Similarly, even for a particular server there are various programs such as in.ftpd or the expanded function version wu.ftpd.

Domain Name Server (BIND)

A name server uses the host name obtained from the IP address or the inverse IP address search to find host names using a distributed database function. One type of host name search program is /etc/hosts. This program searches the internal host from the /etc/hosts list. DNS, requiring that at least one domain be the same, provides host information on the host's internal domain. The `bind` program is the most common way of providing these DNS functions.

Depending on their usage and functions there are four types of name servers. The configuration for the first three of these is discussed under *Name Server Mode Settings* on page 2–11.

- Primary name server

Manages all the host name information for the internal domain. It handles the mail server that routes mail and sends domain information to the secondary server. An important part of its role is to exchange information with other domain name servers.

- Secondary name server

Serving as a back-up to the primary server, it holds a copy of the primary name server's information. In case of trouble with the primary, the secondary name server can take over. In order to connect to the Internet there must be either the primary or secondary name server.

- Cache server

Upon instruction from a client, the cache server can forward requests to the main domain server for action. As the name "cache" indicates this server temporarily caches or stores queries sent to it and is capable of responding to requests rapidly. For large sites this has the merit of reducing the load on the primary and secondary servers.

- Slave server

This stores essentially the same information as the primary server. The difference is the slave server cannot resolve any requests without first issuing a request to the dedicated main server.

BIND Package Contents

The four packages containing the BIND components are found on the *Turbolinux Server 6 for zSeries and S/390 Install CD* under:

- /RPMS/s390/bind-8.2.2P5-3.s390.rpm.
- /RPMS/s390/bind-contrib-8.2.2P5-3.s390.rpm.
- /RPMS/s390/bind-devel-8.2.2P5-3.s390.rpm.
- /RPMS/s390/bind-utils-8.2.2P5-3.s390.rpm.

It is best to install the caching-nameserver package at the same time. Use the `rpm -ql` option to confirm the contents of the package. Enter the command shown below.

```
# rpm -ql bind
```

The command results in the listing shown below.

```
/etc/named.conf
/etc/named/master
/etc/named/named.root
/etc/named/slave
/etc/rc.d/init.d/named
/usr/bin/h2n
/usr/sbin/named
/usr/sbin/named-xfer
/usr/sbin/ndc
/usr/share/man/cat1/h2n
/usr/share/man5/named.conf
/usr/share/man7/hostname
/usr/share/man7/mailaddr-bind
/usr/share/man8/named-xfer
/usr/share/man8/ndc
```

NOTE The contents of the directory /usr/share/doc/packages/bind-8.2.2P5 are omitted from the above return.

The command `rpm -ql caching-nameserver` results in the listing shown below.

```
/etc/named.boot
/etc/named.conf
/var/named/named.ca
/var/named/named.local
```

BIND

The first version of BIND was written at UCB (University of California Berkeley) and named 4.3BSD UNIX. Now it has been ported to Linux, other versions of UNIX, Windows NT, and OS/2.

Up to version 4.8.3 the program was developed under a grant from DARPA at UCB's Computer Systems Research Group (CSRG).

Versions 4.9, 4.91 were released by DEC (now part of Compaq).

Version 4.92 sponsored by Vixie Enterprises.

From version 4.9.3 onward development and maintenance have been handled by ISC (Internet Software Consortium) with new versions coming out one after the other.

Version BIND8 was released in May, 1997.

BIND Configuration File

To configure BIND, edit the resolv.conf file. Open it using the `less` command as follows:

```
# less /etc/resolv.conf
```

Enter the names against their IP addresses.

You may have to edit the nsswitch.conf file. See the `man nsswitch.conf` page for the circumstances under which this would be done. Open it using the `less` command as follows:

```
# less /etc/nsswitch.conf
```

Named Startup Script

The main BIND program contains the daemon named. The startup script for named is /etc/rc.d/init.d/named. Enter the following command to display the contents of this daemon:

```
# less /etc/rc.d/init.d/named
```

An abbreviated version of the result of this command is in the listing shown below:

```
#!/bin/sh
# named
# chkconfig: 345 55 45
./etc/rc.d/init.d/functions
./etc/sysconfig/network
[ $ {NETWORKING} = "no" ] && exit 0
[ -f /usr/sbin/named ] || exit 0
[ -f /etc/named.conf ] || exit 0
case "#1" in
      start)
      # Start daemons.
      echo -n "Starting named: "
      daemon named -u bind
      echo
      touch /var/lock/subsys/named
      ;;
stop)
      # Stop daemons.
      echo -u "Shutting down named: "
      killproc named
      rm -f /var/lock/subsys/named
      echo
```

```
        ;;
    status)
        /usr/sbin/ndc status
        exit $?
        ;;
    restart)
        /usr/sbin/ndc restart
        exit $?
        ;;
    reload)
        /usr/sbin/ndc reload
        exit $?
        ;;
    probe)
        /usr/sbin/ndc reload>/dev/null 2> &1 || echo start
        exit 0
        ;;
    *>
        echo "Usage: named {start|stop|status|restart}"
        exit 1
    esac
    exit 0
```

Starting and Stopping named

Start named by inputting the following command:

```
# /etc/rc.d/init.d/named start
```

To stop named use this command:

```
# /etc/rc.d/init.d/named stop
```

named Operation Confirmation

To confirm that named is running properly input the following command:

```
# ps aux | grep named
root     203  0.0  3.2  1440   1000  ?   S   00:55  0:00   named
```

If the return is not something similar to the response shown above, either named is not running properly or it failed to install. Check whether named is running or reinstall BIND. For more information, see *Managing System Processes* on page 1–25 and *Listing and Installing Packages* on page 1–33.

Name Server Mode Settings

BIND has three operating modes. As mentioned before each mode has certain functions.

1. primary mode
2. secondary mode
3. cache only server mode

The default mode setting in Turbolinux is cache only server. Cache only server mode requires only two files: /etc/named.boot and /etc/resolv.conf.

/etc/named.boot file

```
directory                          /var/named
cache           .                  named.ca
primary         0.0.127.in-addr.arpa   named.local
```

/etc/resolv.conf file

```
domain          example1.com
search          example1.com
nameserver      000.168.0.1
nameserver      000.168.0.2
```

When using the primary or secondary modes it is necessary to change the configuration.

For secondary mode the settings for /etc/named.boot are shown below. In this example the primary server IP address is 000.168.0.2.

/etc/named.boot file

```
directory                                       /etc/named
cache                                           namedroot
secondary   example1.com              000.168.0.2   example1.zone.bak
secondary   0.168.000.in-addr.arpa    000.168.0.2   example1.rev.bak
secondary   0.0.127.in-addr.arpa      local.rev
```

The zone file and reverse file are acquired from the primary server IP address 000.168.0.2 and then a backup file is created. As the configuration for the primary server is rather complicated we will use an example of a connection to the OCN economy internet site to explain. OCN's web site has information on configuring.

The home page of Open Communications Networks Inc. is http:/www.ocn.com/. For more information , see *References for BIND and Resolver* on page 2–24.

Configuration Example (OCN economy example)

Setup has the six files which are shown below. The files /etc/named.boot and /etc/resolv.conf are required, the others are optional. These choices are specified within the /etc/named.boot file.

File Name	Actual File Name (example)
boot file	/etc/named.boot
resolver file	/etc/resolv.conf
cache file	/var/named/named.root
loop-back file	/var/named/0.0.127.in-addr.arpa
normal file	/var/named/example1.com
inverse file	/var/named/2.0.168.000.in-addr.arpa

The following assumptions are used to create an example of a configuration.

IP address	000.168.0.0 ~ 000.168.0.15
subnet mask	255.255.255.240
domain name	example1.com
secondary domain name	203.139.160.69 (host name ns-tk011.ocn.ad.us)
	203.139.161.37 (host name pns.ocn.ad.us)

In the case that a primary name domain is created beforehand, the IP address allotment for OCN is a total of 16. The creation process proceeds as shown in the example below.

ATTENTION Do not use the IP address in the example given here; set the address to the one which applies to your system setup.

IP Address	Host Name	Explanation
000.168.0.0	none	network address (fixed)
000.168.0.1	none	default gateway (root) of fixed bandwidth
000.168.0.2	ns.example1.co.us	name domain (primary) fixed bandwidth
000.168.0.3	unix1.example1.com	UNIX or Linux terminal
000.168.0.4	win1.example1.com	Windows terminal
000.168.0.5	mac1.example1.com	Macintosh terminal
000.168.0.6	(not used in this example)	
000.168.0.7	(not used in this example)	
000.168.0.8	(not used in this example)	
000.168.0.9	(not used in this example)	
000.168.0.10	(not used in this example)	
000.168.0.11	(not used in this example)	
000.168.0.12	(not used in this example)	
000.168.0.13	(not used in this example)	
000.168.0.14	(not used in this example)	
000.168.0.15	none	broadcast address (fixed)

Boot file (/etc/named.boot) Configuration Example

directory		/var/named	[1]
cache	.	named.ca	[2]
primary	0.0.127.in-addr.arpa	0.0.127.in-addr.arpa	[3]
secondary	example1.com	example1.com	[4]
secondary	2.0.168.000.in-addr.arpa	2.0.168.000.in-addr.arpa	[5]

[1] In the directory line the directory is set to the name given in the configuration file. Here it is set to /var/named.

[2] In the cache line the cache file name is specified. Download the most recent file version from ftp://rs.internic.net/domain/named.root. Here we have configured the name to the most recent

file version, named.root. In the primary line the configurations are setup for the loopback file, the normal file (zone file), and the inverse file (reverse file).

[3] The loopback file name is configured here. In this case it is set to 0.0.127.in-addr.arpa.

[4] The normal file (zone file) is configured here. Here it is example1lcom.

[5] The inverse file (reverse file) is configured here. Be sure to set this to the network address to inverse of the normal address, in this case, 2.0.162.000.inaddr.arpa.

Resolver File (/etc/resolv.conf) Configuration Example

```
domain          example1.com    [1]
nameserver      000.168.0.2     [2]
nameserver      000.168.1.2     [3]
nameserver      000.168.2.2     [4]
```

[1] In the domain line the domain name is set, in this case example1.com. The nameserver line configures the name server IP address. At least two addresses must be specified, the primary and secondary addresses.

[2] Sets the primary name server IP address, in this case 000.168.1.2.

[3] Sets the secondary name server IP address. In this case it is set to address provided by service provider, namely 000.168.1.2

[4] Sets the tertiary name server IP address. Tertiary names are not essential. In this case it is set to address provided by service provider, namely 000.168.2.2.

Loop-back File (/var/named/0.0.127.in-addr.arpa) Configuration Example

```
0.0.127.in-addr. IN      SOA    ns.example1.com      root.ns.example1.com (
19990318         ;       Serial                      [1]
10800            ;       Refresh after 3 hours
3600             ;       Retry after 1 hours
604800           ;       Expire after 1 week
86400            ;       Minimum after 1 week

0.0.127.in-addr.arpa.    IN    NS    ns.example1.com.    [2]
0.0.127.in-addr.arpa.    IN    NS    ns.2.provider.ad.   [3]
0.0.127.in-addr.arpa.    IN    NS    ns3.provider.ad.    [4]
1.0.0.127.in-addr.arpa.  IN    PTR   localhost           [5]
```

[1] The configuration serial number value. Any value is acceptable. However when changing the file or reconfiguring be sure to increment the number value.

[2] Sets the primary name server host name, in this case ns.example1.com.

[3] Sets the secondary name server host name, in this case to ns2.provider.ad as assigned by the service provider ns.provider.ad.

[4] Sets the tertiary name server host name, in this case ns3.provide.ad as assigned by the service provider ns.provider.ad.

Normal File (zone file) (/var/named/example1.com) Configuration Example

```
example1.comIN      SOA    ns.example1.com      root.ns.example1.com(
        19990318    ; Serial                    [1]
        10800       ; Refresh after 3 hours
        3600        ; Retry after 1 hours
        604800      ; Expire after 1 week
        86400       ; Minimum TTL of 1 day

    example1.com.   IN     NS    ns.example1.com.    [2]
    example1.com.   IN     NS    ns2.provider.ad.    [3]
```

```
example1.com.       IN      NS      ns3.provider.ad.    [4]
example1.com.       IN      MX      10  ns2.example1.com.[5]
localhost           IN      A       127.0.0.1           [6]
ns.example1.com.    IN      A       000.168.0.2         [7]
unix.example1.com.IN        A       000.169.0.3         [8]
win1.example1.com.IN        A       000.168.04          [9]
mac1.example1.com.IN        A       000.168.05          [10]
mail.example1.com.  IN      CNAME   ns.example1.com.    [11]
www.example1.com.   IN      CNAME   ns.example1.com.    [12]
ftp.example1.com.   IN      CNAME   ns.example1.com.    [13]
```

[1] The configuration serial number value. Any value is acceptable. However when changing the file or reconfiguring be sure to increment the number value.

In the line with IN NS, NS is an abbreviation for Name Server that configures the name server host. [2], [3], and [4] are the primary, secondary and tertiary host names settings. In this case they are ns.example1.com, ns2.example1.com, and ns3.example1.com, respectively.

In the line with IN MX, MX stands for Mail eXchange.

[5] shows the value of the mail server priority order. The smaller the value, the higher the priority. Here the mail server host name is also specified, in this case the primary name server and mail server are the same. Thus the high value of 10 is sets a lower priority for the mail server host ns.example1.com.

In the IN A line, A stands for Address, the host name corresponding to the IP address.

[6] The IP address for the localhost. localhost indicates your own host which is usually set as the 000.0.0.1, the value shown in this example.

[7] The IP address for ns.example1.com, in this case 000.168.0.2.

[8] [9] [10] The IP addresses for each host, in this case, unix1.example1.com, win1.example1.com and mac1.example1.com which correspond to IP addresses 000.168.0.3, 000.168.0.4, and 000.168.0.5, respectively.

In the line with IN CNAME, CNAME stands for Canonical NAME (the standard name), the alias IN CNAME standard naming convention. Sets the alias for the primary server.

[11] [12] [13] The alias names, in this case, the mail server name mail.example1.com, the web server name www.example1.com, and the FTP server name ftp.example1.com.

Inverse File (reverse file) (/var/named/2.0.168.000.in-addr.arpa) Configuration Example

```
2.0.168.000.in-addr.arpa.   IN    SOA    ns.example1.com.

roots.example1.com (
        19990318    ;       Serial                  [1]
        10800       ;       Refresh after 3 hours
        3600        ;       Retry after 1 hours
        604800      ;       Expire after 1 week
        86400       ;       Minimum TTL of 1 day

        IN  NS      ns.example1.com.        [2]
        IN  NS      ns2.provider.ad.        [3]
        IN  NS      ns3.provider.ad.        [4]
        IN  PTR     example1.com.           [5]
        IN  A       255.255.255.240         [6]
2       IN  PTR     ns.example1.com.        [7]
3       IN  PTR     unix1.example1.com.     [8]
4       IN  PTR     win1.example1.com.      [9]
5       IN  PTR     mac1.example1.com.      [10]
```

[1] The configuration serial number value. Any value is acceptable. However when changing the file or reconfiguring be sure to increment the number value. In this case the value is set to the date.

In the line with IN NS, NS is a abbreviation for Name Server that configures the name server host.

[2] [3] [4] Are the primary, secondary, and tertiary host names settings. In this case they are ns.example1.com, ns2.provider.ad, and ns3.provider.ad, respectively.

In the line with IN PTR, PTR stands for PoinTer (pointer).

[5] Sets the corresponding domain name, in this case the domain name example1.com.

In the line with IN A, the subnet mask is set.

[6] In this case, the subnet mask is 255.255.255.240 and that is mask set.

[7] [8] [9] [10] IP address number is attached at the end of each. The PTR (pointer) record indicates the actual host name.

Configuration Check and Confirmation

The proper operation of bind is checked using `ping` and `nslookup`. `Ping` is included in the netkit-base package together with inetd. The `nslookup` program is available in RPM format in the bind-utils package.

First use `ping` to test whether your own domain name in the host is returned to you when a message is sent to the external host. In the example below, your own address 000.168.0.2 is pinged. To stop the ping program use <CTRL> +<c>.

```
> ping 000.168.0.2

PING 000.168.0.2 (000.168.0.2) : 56 data bytes
64 bytes 000.168.0.2: icmp_seq=0 ttl=128 time=0.5 ms
64 bytes 000.168.0.2: icmp_seq=0 ttl=128 time=0.5 ms
64 bytes 000.168.0.2: icmp_seq=0 ttl=128 time=0.5 ms
64 bytes 000.168.0.2: icmp_seq=0 ttl=128 time=0.5 ms
64 bytes 000.168.0.2: icmp_seq=0 ttl=128 time=0.5 ms
--- 000.168.0.2 ping statistics ---
```

```
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.5/0.5/0.5 ms
```

If you seen a display similar to the one above, your network is working properly. Check the internal domain host and external host connection in the same way.

Checking using nslookup is performed in the following manner.

(1) First, change your own IP address to the host name. Check that the host name is properly changed to the IP address.

(2) Using the same method, confirm that the internal domain host name is changed.

(3) Last, check that the external (internet) host is properly changed.

Enter the nslookup command.

```
# nslookup

Default Server:  ns.example1.com
Address:   000.168.0.2
Aliases:     2.0.168.000.in-addr.arpa
```

This is what the displayed output should look like. When you first start nslookup it may take a while to finish running.

Next using your own or the domain internal host name to a test to see if you can retrieve the IP address information.

As in the following example, if you are able to lookup this type of information, you system is working properly.

Example using a dummy address:

```
> ns.example1.com
Server: ns.example1.com
Address: 000.168.000.in-addr.arpa
Aliases: 2.0.168.000.in-addr.arpa
Name: ns.example1.com
Address: 000.168.0.2
>Address: 000.168.0.2
```

```
Server: ns.example1.com
Address: 000.168.0.2
Aliases: 2.0.168.000.in-addr.arpa
Name:   ns.example1.com
Address:   000.168.0.2
Aliases: 2.0.168.000.in-addr.arpa
```

Example using a dummy internal domain host:

```
> unix1.example1.com
Server: ns.example1.com
Address: 000.168.0.2
Aliases: 3.0.168.000.in-addr.arpa
Name: unix1.example1.com
Address: 000.168.0.3
>Address: 000.168.0.3
Server: ns.example1.com
Address: 000.168.0.2
Aliases: 3.0.168.000.in-addr.arpa
Name:   ns.example1.com
Address:   000.168.0.2
Aliases: 000.168.0.2.in-addr.arpa
```

Example using an external host:

```
> blue.ocn.ne.us
Server: ns.example1.com
Address: 000.168.0.2
Aliases: 226.162.139.203.in-addr.arpa
Non-authoritative answer:
Name: blue.ocn.ne.us
Address: 203.139.160.87
>203.139.160.87
Server: ns.example1.com
Address: 000.168.0.2
Aliases: 226.162.139.203.in-addr.arpa
Name:   blue.ocn.ne.us
Address:   203.139.160.87
```

Resolver Program

The `resolver` receives name requests from the client and forwards the request to the name server and returns answers to the client. While the name server is the program that handles making decisions for the server program, the `resolver` is the client side program for the same function.

`Resolver` is a component of the bind-utils package. Use the `rpm -ql` option to confirm the contents of the package. Enter the command shown below.

```
# rpm -ql bind-utils
```

The command results in the listing shown below:

```
/usr/bin/addr
/usr/bin/dig/
/usr/bin/dnsquery
/usr/bin/host
/usr/bin/mkservdb
/usr/bin/nslookup
/usr/bin/nsupdate
/usr/lib/nslookup.help
/usr/sbin/dnskeygen
/usr/sbin/irpd
```

NOTE The relevant contents of /usr/share/man are listed under *References for BIND and Resolver* on page 2–24.

Resolver configuration files

The two resolver configuration files are /etc/host.conf and /etc/resolv.conf.

The /etc/host.conf handles name inquiries issued from the name server by either name server or by the traditional UNIX hosts file method.

With the host file name service all hosts must be stored in the same hosts file. For a large network this makes system maintenance time-consuming and unrealistic. But the merit of name server is in the load reduction that makes for good system response when running a small network with only or four or five hosts.

To display the /etc/host.conf file, enter the command below

```
# less /etc/host.conf
```

This command returns the following contents.

```
order hosts,bind
multi on
```

Below is the process for running the `order` program. In this case, first test the `name resolution` function. If it does not resolve, it means you are running `bind`. The multi on message means that if the name resolution results in multiple addresses they will be processed. With multi off only the first address found will be used in a response.

The file /etc/resolv.conf is for use by the domain name and name server. Display this configuration file using the `less` command as shown below:

```
# less /etc/resolv.conf
```

A typical example of the contents of /etc/resolv.conf is shown below.

```
domain example1.com
search example1.com
nameserver 000.168.0.2
```

The word `domain` here denotes the domain associated with the server. The word `search` denotes the auxiliary domain to be used when no domain is returned by the host inquiry. `Nameserver` is the name server program.

References for BIND and Resolver

- Bind (Berkeley Internet Name Domain)
  http://www.isc.org/bind.html/
- DEC (Digital Equipment Corporation
  http://www.dec.com/
- Compaq/Digital Equipment
  http://www.unix.digital.com/faqs/publications/base_doc/
- ISC (Internet Software Consortium)
  http://www.isc.org/
- The home page of OCN Inc.,
  http://www.ocn.com/
- The following man pages are available for `bind`:

  `hostname` `mailaddr-bind` `named` `named.conf`
  `ndc` `named-xfer`

- The following man pages are available for `bind-utils`:

  `dig` `dnskeygen` `dnsquery` `host`
  `irs.conf` `resolv.conf` `resolver` `named-bootconf`
  `nslookup`

Firewall Maintenance (ipchains)

Ipchains is used to set up, maintain, and inspect the firewall rules in the Linux kernel.

Connecting Linux to the internet, building a firewall to protect your system from intruders, and enabling network address translation for a LAN are frequent sources of questions.

All traffic through a network is sent in the form of packets. For example, downloading a 50 kilobyte package might cause you to receive 36 or so packets of 1460 bytes each.

The first part of each packet indicates its source and destination, the type of the packet, and other administrative details. This part of the packet is called the header. The rest of the packet, containing the actual data being transmitted, is usually called the body.

A packet filter is a piece of software that looks at the headers of packets as they pass through, and decides the fate of the entire packet. It might decide to deny the packet (i.e. discard the packet as if it had never received it), accept the packet (i.e. let the packet go through), or reject the packet (like deny, but tells the source of the packet that it has done so).

Linux ipchains is required to administer the IP packet filters in Linux kernel versions 2.1.102 and above. Under Linux, packet filtering is built into the kernel. Turbolinux Server 6 for zSeries and S/390 meets this criteria.

Linux ipchains is a rewrite of the Linux IPv4 firewalling code and a rewrite of `ipfwad`.

A firewall is a device that protects a private network from the internet as a whole. The firewall computer can reach both the protected network and the Internet. The protected network cannot

reach the Internet, and the Internet cannot reach the protected network.

There are two types of firewalls: IP or Filtering Firewalls that block all but selected network traffic, and Proxy Servers that make the network connections for you.

To get started quickly, we suggest you read about ipchains and firewall configuration in the following documents:

- man ipchains
- /usr/share/doc/packages/ipchains-1.3.9/HOWTO.txt.gz.

To read the ipchains document, use the command:

```
# zless /usr/share/doc/packages/ipchains-1.3.9/HOWTO.txt.gz
```

The package containing ipchains is found on the *Turbolinux Server 6 for zSeries and S/390 Install CD* under /RPMS/s390/ipchains-1.3.9-3.s390.rpm.

Name	**ipchains**
configuration file	/etc/sysconfig/ipchains.rules
startup script	/etc/rc.d/init.d/ipchains
related files	/proc/net/ip_fwchains
	/proc/net/ip_masquerade

ipchains Package Contents:

To view the contents of the ipchains package use the rpm -ql option as shown below

```
# rpm -ql ipchains
```

The command results in the listing shown below:

```
/etc/rc.d/init.d/ipchains
/etc/sysconfig/ipchains.rules
/sbin/ipchains
/sbin/ipchains-restore
/sbin/ipchains-save
/sbin/ipfwadm
/sbin/ipfwadm-wrapper
```

NOTE The contents of /usr/share/doc/packages/ipchains-1.3.9 are omitted. The contents of /usr/share/man are listed in *References for ipchains and firewall* on page 2–28.

ipchains Configuration File (/etc/sysconfig/ipchains.rules)

Configuration of Turbolinux Server firewall rules should take place during network configuration and setup at system boot time.

At startup ipchains reads configuration information from a file which by default is the configuration file /etc/ipchains.rules.

Turbolinux Server 6 for zSeries and S/390 uses the firewall startup scripting process based on /etc/rc.d/init.d/ipchains. At system startup this script is called with the 'start' argument. This causes the script to load whatever firewall rules are present in the rules table /etc/sysconfig/ipchains.rules.

Administrators may enable or disable the firewall rules by executing at any time the following from a command line.

To enable firewall rules, type:

```
sh /etc/rc.d/init.d/ipchains   start
```

To disable firewall rules, type:

```
sh /etc/rc.d/init.d/ipchains   stop
```

Preparing Firewall Rules. The preparation of the ipchains.rules table can be accomplished by first loading firewall rules, either manually or using the GUI firewall control center, GFCC, and then by saving the rules by executing the following:

```
ipchains-save  > /etc/sysconfig/ipchains.rules
```

Please refer to the documentation for ipchains for information regarding how to create firewall rules at /usr/share/doc/packages/ipchains-1.3.9.

A rules table can be prepared on one machine. For example, a system that has been installed as an intranet server, thus with the resources needed to use the GFCC program to create the firewall rules table, can be copied to a basic firewall system for use.

References for ipchains and firewall

To learn the basics of firewall systems and to obtain some details on setting up both a filtering and proxy firewall on a Linux based PC, use:

- The Firewall-HOWTO, available at http://metalab.unc.edu/LDP/HOWTO/Firewall-HOWTO.html/
- The official page is the Linux IP Firewall Chains Page at http://www.rustcorp.com/linux/ipchains/
- The following man pages are available for ipchains and firewall:

```
ipchains            ipchains-restore    ipchains-save
ftpshut             ipfwadm-wrapper
```

File Transfer Protocol Client (TFTP)

TFTP (part of the InetUtils package) 1.3.2, is the user interface to the Internet TFTP (Trivial File Transfer Protocol), which allows users to transfer files to and from a remote machine. The remote host may be specified on the command line, in which case tftp uses host as the default host for future transfers.

TFTP has been implemented on top of the Internet User Datagram protocol (UDP or Datagram) so it may be used to move files between machines on different networks implementing UDP. (This should not exclude the possibility of implementing TFTP on top of other datagram protocols.) It is designed to be small and easy to implement. Therefore, it lacks most of the features of a regular FTP. The only thing it can do is read and write files (or mail) from and to a remote server. It cannot list directories, and currently has no provisions for user authentication. In common with other Internet protocols, it passes 8-bit bytes of data.

InetUtils is a collection of common network programs. Among other features, it includes:

- An ftp client and server.
- A telnet client and server.
- An rsh client and server.
- An rlogin client and server.
- A tftp client and server.

ATTENTION It is not safe to install TFTP on any server that is open to the Internet.

TFTP Configuration

In order to use TFTP, open the file /etc/inetd.conf and delete the hash symbol (#) preceding the following line:

```
#ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -1 -a
```

TFTP Reference

- The TFTP package contains the following man pages:

```
tftp            in.tftpd        tftpd
```

File Transfer Protocol Server (ProFTP)

ProFTP is an ftp client and server developed by Labtam which allows the user to transfer files between a PC and a remote computer using the ARPANET standard File Transfer Protocol. The program can transfer files in two different format types: ASCII format for text files, and Binary format for image files.

FTP must run on a machine that is configured for TCP/IP network communication or Internet access. Access can be gained through Winsock and the use of any dial-up provider, a Remote Access Server (RAS), or a direct connection via a local area network that supports TCP/IP. In order for FTP to communicate with a remote computer, that computer must have a server implementation of FTP (FTP server based on TCP/IP transports).

You can be connected to only one remote computer at a time during an FTP session. However, you can run multiple FTP sessions (FTP clients) simultaneously in separate windows, with each session connected to a different host.

ProFTP is found on the *Turbolinux Server 6 for zSeries and S/390 Install CD* under /RPMS/s390/proftpd-1.2.0rc2-1.s390.rpm.

Name	**ProFTP**
daemon	/usr/sbin/proftpd
configuration file	/etc/proftpd/proftpd.conf
startup script	/etc/rc.d/init.d/inet

ProFTP Package Contents

To view the contents of the ProFTP package use the `rpm -ql` option as shown below.

```
# rpm -ql proftpd
```

The following is an abbreviated listing of the results:

```
/etc/pam.d/proftp
/etc/proftpd/ftpusers
/etc/proftpd/proftp.conf
/usr/bin/ftpcount
/usr/bin/ftpwho
/usr/bin/ftpshut
/usr/bin/in.proftpd
/usr/bin/proftpd
```

NOTE The online manual files and the contents of the /usr/share/doc/packages/proftpd-1.2.0pre10 folder are omitted from this list.

ProFTP Startup Script and Options

The ProFTP startup script is located in /etc/rc.d/init.d/inet. Use the following command to access it:

```
# /etc/rc.d/init.d/inet [start|stop|restart]
```

ProFTP Configuration File (/etc/proftpd/proftpd.conf)

To read the very well documented ProFTP configuration file, enter the following command:

```
# less /etc/proftpd/proftpd.conf
```

The following is an abbreviated listing of the result:

```
ServerName"Test Installation"
ServerTypeinetd
DefaultServeron
ServerIdentoff
Port21
Umask022
MaskInstances30
PAMConfigProFTP
AuthPAMAuthoritativeOn
Usernobody
Groupnobody
<Directory/*>
AllowOverwriteOn
</Directory>
```

ProFTP Operation Confirmation

To confirm that ProFTP is running properly enter the following command:

```
# ps aux | grep proftp
root     3930 0.0 1.3 1140   404 tty1 S 13:17 0:00 grep proftp
```

If the return is not something similar to the response shown above, either `proftp` is not running properly or it failed to install. Check whether ProFTP is running or try reinstalling it.

References for ProFTP

- An extensive user guide is available at http://hamster.wibble.org/proftpd/userguide/linked/userguide.html/
- http://www.lab-pro.com/index.cfm/
- http://www.proftpd.net/

- The following man pages are available for ProFTP:

  ```
  proftpd        ftpcount       ftpwho         xferlog
  ftpshut
  ```

http Server (Apache)

Apache is the most widely used web server in the world. According to February, 1999 data, Apache holds over 50% of the world share for web servers and has become the de facto standard.

Apache Project

Apache's collaborative development group is known as the Apache Project. Apache features an open source code, freeware with numerous features, high efficiency, a stable environment and easy system management.

Information on Apache is available at the developers' Apache Project web site. In addition, the web magazine *Apache Week* gathers together a myriad of useful information. For more information, see *References for Apache and httpd* on page 2–43.

Apache is found on the *Turbolinux Server 6 for zSeries and S/390 Install CD* under /RPMS/s390/apache-1.3.14-1.s390.rpm.

Name	**Apache**
daemon	/usr/sbin/httpd
configuration file	/etc/httpd.conf (and three others listed on page 2–36)
startup script	/etc/rc.d/init.d/httpd
log file	/var/lock/subsys/inet

httpd Startup Script

Apache includes the daemon called httpd for which the startup script is /etc/rc.d/init.d/httpd. To display the startup script, enter the following command:

```
# less /etc/rc.d/init.d/httpd
```

The daemon, httpd, has three main options:

```
/etc/rc.d/init.d/httpd [start|stop|restart]
```

After making any changes to the configuration you will need to restart httpd to enable the changes. To restart httpd, use this command:

```
# /etc/rc.d/init.d/httpd  restart
```

Confirm that httpd is operating properly by using this command:

```
# ps aux  |  grep  httpd
```

If issuing the above confirmation command does not result in a display like:

```
nobody  351  0.0 3.7  1916  1164  ?  S  08:13  0:00  httpd
```

it is possible that httpd is not operating or that the installation failed.

Verify that the package is installed by using the command:

```
# rpm -ql apache
```

Restart httpd by using the command:

```
# /etc/rc.d/init.d/httpd restart
```

httpd Configuration File

Httpd has four associated configuration files. The file location and main functions are listed below.

/etc/httpd/conf/httpd.conf	overall configuration settings for the httpd daemons
/etc/httpd/conf/srm.conf	detailed settings for related html documents
/etc/httpd/conf/access.conf	access operation configuration
/etc/mime.types	listing of MIME file types

To utilize Apache's wide range of features, you must configure the above files. Detailing all of Apache's many features would take several volumes; below we explain the basic features.

httpd.conf

The httpd.conf file is for the overall configuration setting for the httpd daemons. Confirm the contents of httpd.conf using the following command:

```
# less  /etc/httpd/conf/httpd.conf
```

This should return the following display:

```
ServerType standalone
Port 80
HostnameLookups off
User nobody
Group nobody
ServerAdmin webmaster@example.com
ServerRoot /etc/httpd
ErrorLog logs/error_log
LogLevel warn
LogFormat “%h %l %u %t  “%r” %>s %b” common
CustomLog logs/access_log common
```

NOTE Lines commented out with "#" are omitted

The basic directives are:

- ServerType

Directives can change the startup mode of httpd to either standalone or inetd mode. The default is standalone mode. Httpd can start in standalone mode or super server mode but unless you have a special reason there are strong arguments for starting in the standalone mode. Apache sets standalone as the default.

- Port

Httpd sets the port number at which client requests for connection are received. The default value is 80. Port 80 is widely known to the computing public as the client port for the httpd server. This is also the httpd server's port used by the WKS (Well Known Services) record.

Unless you have a special reason, such as wishing to display only selected users or to start httpd from a directory other than the root, it is recommended that you maintain the default value.

- HostnameLookups

Determines whether the access log file of client requests records the host name or the IP address. The default of OFF records IP addresses. This is the recommended setting.

- `User` and `Group`

Sets the user and group names of the httpd process owner. The defaults are nobody for both user and group. The httpd process, that is the home page, intentionally sets the security level very low by using nobody, on the assumption that many and unspecified users will use the system.

Unless you have a special reason, maintain the default setting. If you should change the setting, be particularly careful about specifying the user and group settings for the selected users.

- ServerAdmin

Sets the mail address for the httpd manager. The default is `root@localhost`. A message from the manager can be added here. The default setting is fine; however if there are several web managers it is convenient to substitute the address of a web manager mail list instead. Note the above example uses webmaster@example.com.

- ServerRoot

The location of the configuration files is set by default to /etc/httpd. This is the recommended location.

- ErrorLog

The name for the log for recording errors. By default error messages are saved in the file logs/error_log. If you are using virtual hosts separate error logs can be defined for each virtual host.

- LogFormat and CustomLog

`LogFormat` specifies the format of access log files. `CustomLog` does the same for access logs. If you are using virtual hosts, separate logs can be defined for each virtual host.

srm.conf

Confirm the contents of the srm.conf file with the following command:

```
# less  /etc/httpd/conf/srm.conf
```

A display similar to that below should appear (lines commented out with # are omitted)

```
DocumentRoot /home/httpd/html
UserDir public_html
DirectoryIndex index.html index.shtml index.cgi index.php3
FancyIndexing on
ReadmeName README
HeaderName HEADER
IndexIgnore.??* *~ *# HEADER* README* RCS
AccessFileName .htaccess
TypesConfig /etc/mime.types
Alias /icons/ /home/httpd/icons/
ScriptAlias /cgi-bin/ /home/httpd/cgi-bin/
```

The basic directives of srm.conf are:

- DocumentRoot: Specifies the location of HTML files.
- UserDir: Specifies the public user directory.
- DirectoryIndex: By default, this is set to display file names.
- Alias: Specifies the directory alias names for names not listed in DocumentRoot.
- ScriptAlias: Specifies the directory alias names for CGI scripts and other executable files.

access.conf

Confirm the contents of the access.conf file with the following command:

```
# less  /etc/httpd/conf/access.conf
```

A display similar to that below should appear (lines commented out with # are omitted).

```
<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>
<Directory /home/httpd/html>
Options Indexes FollowSymLinks
AllowOverride None
order allow,deny
allow from all
</Directory>
<Directory /home/httpd/cgi-bin>
AllowOverride None
Options ExecCGI
</Directory>
```

The format of the access.conf file is similar to the HTML format standard. The lines from `<Directory filename>` to `</Directory>`, i.e., the first four lines of the above listing, configure the directory. By default the directories are set to the root (/) directory, HTML directory (/home/httpd.html), and cgi-bin directory (/home/httpd/cgi-bin). Other directories can be added.

Options. The main options are as follows:

None	disables all options
All	enables all options
Indexes	when the DirectoryIndex srm.conf is not specified, this displays a list of directories

ExecCGI	enables running CGI files
Includes	enables running SSI (Server Side Include)
IncludesNOEXEC	enables running SSI, but with the `#exec` command will run CGI and with the `#include` command disables running.
FollowSymLinks	enables symbolic linking
SymLinksIfOwnerMatch	enables symbolic linking only when explicitly allowed by the process owner.

AllowOverride. This is the access control file. Designates process when .htaccess is present. The available options are shown below.

None	disables all .htaccess options
All	enables all .htaccess options
AuthConfig	enables only those settings explicitly set in .htaccess
FileInfo	enables only those formats explicitly set in .htaccess
Indexes	enables only those lists displays explicitly set in .htaccess
Limit	enables only those access operations explicitly set in .htaccess
Options	enables only those target directories explicitly defined in .htaccess

order	Specifies the priority of permission granting/denial for access. The available options are shown below.
allow,deny	order of decision process is allow, then deny
deny,allow	order of decision process is deny, allow
mutual-failure	grants permission only when both allow and deny conditions are satisfied

AllowFrom and DenyFrom. Specifies access permission and access denial. Available options are shown below.

all	grants or denies permission to all
domain name	permits access from the specific domain name
IP address	permits access from the specific IP address

Apache Operation Confirmation

To confirm that Apache is running properly, enter the following command.

```
# ps aux | grep apache
```

If the return is not something similar to the response shown below, either httpd is not running properly or it failed to install. Check whether httpd is running or try reinstalling Apache.

```
root     3930 0.0 1.3 1140   404 tty1 S 13:17 0:00 grep apache
```

For more information, see *Managing System Processes* on page 1–25 and *Listing and Installing Packages* on page 1–33.

References for Apache and httpd

You can find more information at the following sites:

- Apache HTTP Server Project
  http://www.apache.org/
- Apache Project: About Apache
  http://www.apache.org/ABOUT_APACHE.html/
- Apache Week
  http://www.apacheweek.com/

The latest information and programs are available for viewing and downloading via the Internet. Source code from the Apache Project development effort is available at their download site or at these mirror sites:

- Apache Project download site,
  http://www.apache.org/dist/
- Apache Project mirror site,
  http://www.apache.org/dyn/closer.cgi/
- The Netcraft Web Server Survey,
  http://www.netcraft.com/survey/

The man page for this application is at:

- `man httpd`
- A well documented online manual is available on an installed Turbolinux system under /home/httpd/html/manual

Lightweight Directory Access Protocol (OpenLDAP)

OpenLDAP is an open-standard directory service protocol designed to handle account information services. The protocol runs over Internet transport protocols, such as TCP, and can be used to access stand-alone directory servers or X.500 directories. The LDAP directory service is based on a client-server model.

There are many different ways to provide a directory service. Different methods allow different kinds of information to be stored in the directory; place different requirements on how that information can be referenced, queried and updated; how it is protected from unauthorized access, etc. Some directory services are local, providing service to a restricted context (for example, the `finger` service on a single machine). Other services are global, providing service to a much broader context.

OpenLDAP Package Contents

To view OpenLDAP's contents, use the `rpm -ql` option:

```
# rpm -ql openldap
```

The command results in the listing shown below.

```
/etc/ldap/defaultbase.ldap
/etc/ldap/ldapfilter.conf
/etc/ldap/ldapsearchprefs.conf
/etc/ldap/ldapserver
/etc/ldap/ldaptemplates.conf
/usr/bin/ldapadd
/usr/bin/ldapdelete
/usr/bin/ldapmodify
/usr/bin/ldapmodrdn
/usr/bin/ldapsearch
/usr/bin/ud
/usr/sbin/xrpcomp
/usr/share/doc/packages/openldap-1.2.10
```

NOTE The contents of the man pages follow the /openldap-1.2.10 files; the contents of both are omitted from this list.

ldap Operation Confirmation

Ldap's operation can be checked using the ps command as shown below:

```
ps aux | grep ldap
```

If ldap is operating properly, the above command should return the message shown below. Check that the program name ldap appears at the far right of the listing.

```
root  2888 0.0  1.3 1140 404 tty1 S  15:51 0:00 grep ldap
```

If you do not see something similar to the message shown above, either ldap is not operating properly or it was not installed. Restart or reinstall it. For more information, see *Managing System Processes* on page 1–25 and *Listing and Installing Packages* on page 1–33.

ldap Startup Script

Below is at a specific example of the default startup script located in /etc/rc.d/init.d/ldap.

To start ldap use the following command:

```
# /etc/rc.d/init.d/ldap start
```

To stop ldap use this command:

```
# /etc/rc.d/init.d/ldap stop
```

ldap Configuration File (/etc/ldap.conf)

To read the very well documented openLDAP configuration file, enter the following command

```
# less /etc/ldap.conf
```

The following is an abbreviated listing of the result:

```
./etc/rc.d/init.d/functions
    ./etc/sysconfig/network
    [ ${NETWORKING} = "no" ] && exit 0
    [ -f /usr/bin/slapd ] || exit 0
    [ -f /usr/bin/slurpd ] || exit 0

    RETVAL=0

    case "$1" in
          start
          echo -n "Starting ldap: "
          daemon slapd
          RETVAL=$?
          if [ $RETVAL -eq 0 ]; then
          if grep -q "^replogfile" /etc/ldap/slapd.conf; then
                daemon slurpd
                RETVAL=$?
          [ $RETVAL -eq 0 ] && pidof slurpd \ cut -f 1 -d " " /
    var/run/slurpd
                fi
          fi
          echo
          [ $RETVAL -eq 0 ] && touch /var/lock/subsys/ldap
          ;;
    stop)
          echo -n "Shutting down ldapf: "
          killproc slapd
          RETVAL=$?
          if [ $RETVAL -eq 0 ]; then
          if grep -q "^replogfile" /etc/ldap/slapd.conf; then
                killproc slurpd
                RETVAL=$?
          fi
    fi
    echo
    if [ $RETVAL -eq 0 ]; then
          rm -f /var/lock/subsys/ldap
          rm -f /var/run/slapd.args
    fi
    ;;
    status)
```

```
        status slapd
        RETVAL=$?
        if [ $RETVAL -eq 0 ]; then
        if grep -q "^replogfile" /etc/ldap/slapd.conf; then
              status slurpd
              RETVAL=$?
        fi
        ;;
restart)
        $0 stop
        $0 start
        RETVAL=$?
        ;;
reload)
        killproc -HUP slapd
        RETVAL=$?
        if [ $RETVAL -eq 0 ]; then
        if grep -q "^replogfile" /etc/ldap/slapd.conf; then
              killproc -HUP slurpd
              RETVAL=$?
        fi
fi
;;
*)
        echo "Usage: $0 start|stop|restart|status}"
        exit 1
esac

exit $RETVAL
```

OpenLDAP also has these configuration files:

```
stopd.conf          stopd.at.conf     stopd.oc.conf
ldaptemplates.conf  ldapfilter.conf   ldapsearchprefs.conf
ldapserver.conf     defaultbase.ldap
```

References for OpenLDAP

- OpenLDAP
  http://www.openldap.org/
- University of Michigan LDAP,
  http://www.umich.edu/~dirsvcs/ldap/index.html/ and
  http://www.umich.edu/~dirsvcs/ldap/doc/
- Manually Implementing Roaming Access,
  http://help.netscape.com/products/client/communicator/manual_roaming2.html/
- Customizing LDAP Settings for Communicator 4.5,
  http://developer.netscape.com/docs/manuals/communicator/ldap45.htm/
- OpenLDAP has eleven man pages listed in the table below:

`ldapadd`	`ldapdelete`	`ldapmodrdn`
`ud`	`ldapfilter.conf`	`ldapsearchprefs.conf`
`ldapsearch`	`ldapmodify`	`ldaptemplates.conf`
`ud.conf`	`ldapfriendly`	

Network File System (NFS)

NFS (Network File System) makes it possible to share files between hosts that are connected to a network. Supported by most installations of Unix, it is an extremely convenient way to transfer files on Unix. Also, with some additional software, you can support NFS on operating systems other than Unix and transparently transfer files between many computers.

On Turbolinux Server 6 for zSeries and S/390, you can install knfsd, which boosts performance over an NFS server that is traditionally run at the user level.

To use NFS, you must configure NFS on both the server side and the user side. On the server side, the edits are made to the basic configuration file, /etc/exports, and then run the daemon. On the client side, you must mount the directories that are being made public.

NFS-server Structure

Confirm the contents of the nfs-server file by using the `rpm -ql` option. Enter the following command:

```
# rpm -ql knfsd
```

The command results in the listing shown below.

```
/etc/rc.d/init.d/nfs
/sbin/rpcdebug
/usr/man/man5/exports.5
/usr/man/man8/exportfs.8
/usr/man/man8/nfsstat.8
/usr/sbin/rpc.mountd
/usr/sbin/rpc.nfsd
/usr/sbin/rpc.rquotad
/usr/man/man8/rquotad.8
/usr/sbin/exportfs
/usr/sbin/nhfsstone
```

```
/usr/sbin/rpc.statd
/var/lib/nfs
/var/lib/nfs/etab
/var/lib/nfs/rmtab
/var/lib/nfs/xtab
```

NOTE This list excludes the contents of /usr/share/doc/packages/knfsd-1.4.7.

All the files listed with the above command are installed when the `nfs-server` RPM package is installed.

To check the operation of rpc.nfsd, enter the following command:

```
# ps ax | grep nfs
4267          pts/0         S         0:00         grep nfs
```

To check the operation of rpc.mount, enter the following command:

```
# ps ax | grep mount
4269        pts/0         S         0:00        grep mount
```

Starting and Stopping NFS

Before running knfsd first check that portmap is running. Run portmap using the following command:

```
# /etc/rc.d/init.d/portmap.init
```

The NFS startup script is /etc/rc.d/init.d/nfs. Startup script options are: `start`, `stop`, and `restart`.

Once you change the settings of NFS, you must restart it in order for the changes to go into effect.

The syntax of the NFS startup script is:

```
#/etc/rc.d/init.d/nfs {start|stop|restart|status}
```

Use the `status` option to check the current status of NFS. If NFS is running, the return will resemble the following:

```
root   766 0.2 0.2 1764 732 ? S 08:13 0:00 nfsd
```

If you do not see this return, NFS may not be running or not installed. You will have to start, restart, or install NFS.

For more information, see *Managing System Processes* on page 1–25 and *Listing and Installing Packages* on page 1–33.

Display the contents of the NFS startup script with this command:

```
# less  /etc/rc.d/init.d/nfs
```

The return of this command looks something like this:

```
#!/bin/sh
#
# nfs This shell script takes care of starting and stopping
the NFS
# services.
# chkconfig: 345 85 20
# description: NFS is a popular protocol for file sharing
across TCP/IP
#             networks. This service provides NFS server func-
tionality, \
#              which is configured via the /etc/exports file.
# probe: true
# Source function library.
. /etc/rc.d/init.d/functions
# Source networking configuration.
if [ ! -f /etc/sysconfig/network ]; then
    exit 0
fi
. /etc/sysconfig/network
# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0
[ -x /usr/sbin/rpc.nfsd ] || exit 0
[ -x /usr/sbin/rpc.mountd ] || exit 0
[ -x /etc/exports ] || exit 0

# Number of servers to be started up by default
RPCNFSDCOUNT=8
```

```
# See how we were called.
case "$1" in
 start)
        # Start daemons.
        echo "Starting NFS services: "
        /usr/sbin/exportfs -r
        echo -n "   Starting NFS statd: "
        daemon rpc.statd
        echo
        echo -n "   Starting NFS quotas: "
        daemon rpc.rquotad
        echo
        echo -n "   Starting NFS mountd: "
        daemon rpc.mountd
        echo
        echo -n "Starting NFS daemon: "
         daemon rpc.nfsd $RPCNFSDCOUNT
        echo
        touch /var/lock/subsys/nfs
        ;;


 stop)
        # Stop daemons.
        echo -n "Shutting down NFS services: "
        /usr/sbin/exportfs -au
        echo -n "Shutting down NFS mountd: "
        killproc rpc.mountd
        echo
        echo -n "Shutting down NFS daemon: "
        killproc nfsd
        echo
        echo -n "Shutting down NFS quotas: "
        killproc rpc.rquotad
        echo
        echo -n "Shutting down NFS statd: "
        killproc rpc.statd
        echo
        rm -f /var/lock/subsys/nfs
        ;;

  status)
        status rpc.statd
```

```
        status rpc.mountd
        status nfsd
        status rpc.rquotad
        ;;
 restart)
        echo -n "Restarting NFS services: "
        echo -n "rpc.statd "
        killall -HUP rpc.statd
        echo -n "nfsd "
        killall -HUP nfsd
        echo -n "rpc.mountd "
        killall -HUP rpc.mountd
        echo -n "rpc.quotad "
        killall -HUP rpc.rquotad
        touch /var/lock/subsys/nfs
        echo "done."
        ;;
 reload)
        /usr/sbin/exportfs
        touch /var/lock/subsys/nfs
        ;;
 probe)
        if [ ! -f /var/lock/subsys/nfs ] ; then
          echo start; exit 0
        fi
        /sbin/pidof rpc.mountd >/dev/null 2>&1; MOUNTD="$?"
        /sbin/pidof nfsd >/dev/null 2>&1; NFSD="$?"
        if [ $MOUNTD = 1 -o $NFSD = 1 ] ; then
          echo restart; exit 0
        fi
        if [ /etc/exports -nt /var/lock/subsys/nfs ] ; then
          echo reload; exit 0
        fi
        ;;
 *)
        echo "Usage: nfs {start|stop|status|restart|reload}"
        exit 1
esac
exit 0
```

NFS Server Settings

The file in which you specify the hosts, users, and public directories that are to given access; the access permissions, etc. is /etc/exports. Each line in /etc/exports uses the following basic format:

```
[directory name]  [host name (options)]
```

Here `directory name` refers to the name of the directory that you want to export, with the full Unix path. The `host name` can be written in either the FQDN (Fully Qualified Domain Name) or IP address format. There are many options; only the main ones are described here.

For details run `man exports`.

Here is a list of the options:

roFile	permissions set to read only.
rwFile	permissions set to read and write.
root_squash	root access from the client is mapped to anonymous (nobody).
no_root_squash	root access from the client is permitted as root.
all_squash	All access is treated as access from nobody.
anonuid=uid	With the root_squash or all_squash options, maps to the anonymous user ID.
anongid=gid	With the root_squash or all_squash options, maps to the anonymous group ID.

NOTE By separating options with a comma, several options can be specified at one time, but if a space is inserted between the option and the command, you may not get the desired results.

An example listing of /etc/exports.

```
/usr            *.turbolinux.com(ro)
/home/you       test (rw,all_squash, anonuid=150,anongid=100)
/home/samba     (ro,all_squash)
```

In the first line, all machines under the turbolinux.com domain are allowed to read (but not write) everything under /usr.

In the second line, the machine called `test` is allowed to read and write to /home/you. Regardless of user, access is given as uid (user ID) set to 150 and gid (group ID) set to 100.

In the third line, note that no host name is specified. This means that all hosts are allowed to read (but not write) /home/samba. All access is done through the nobody account.

Running the Server

Before running the NFS server, check to see if portmap is running with the appropriate settings.

Like other servers, NFS is started with the init script. Because NFS is not set to run by default in Turbolinux Server 6 for zSeries and S/390, you must use `chkconfig` to configure it so that it will run as a daemon after `NFS` is restarted.

```
#chkconfig --add nfs
```

NFS Server Run Check

Use `exportfs` to check the status of NFS exports. (There are other possible uses for the `exportfs` command. Run `man exportfs` for details.)

With the example configuration of /etc/exports above, exportfs looks like this:

```
#exportfs
/usr                    *.turbolinux.com
/home/your              test
/home/samba             <world>
```

The showmount command shows which clients have which directories mounted. For example, if host test2.turbolinux.com has /usr mounted, it would look like this:

```
# showmount -a
All mount points on cadiz.calleprivada:
test2.turbolinux.com:/usr
```

Client Side Settings

NFS servers may have a great number and variety of clients but here we restrict our explanation to Turbolinux.

Basically, the client can use the mount command to mount directories that have been exported by the NFS server.

As an example, try to mount the /usr directory of the NFS server (nfssvr) on /mnt/usr, thus:

```
#mount -t nfs nfssvr:/usr  /mnt/usr/
```

First, you must have created the mount point directory.

By editing the /etc/fstab file, you can use mount without giving all the command line options each time.

For example add the following line to your /etc/fstab file.

```
nfssvr:/usr /mnt/usr  nfs  noauto,rw
```

Given the above, mount can be done with this alone:

```
# mount /mnt/usr
```

This is an extremely simple `mount` example. In actual practice, many options can be given to `mount` on the command line or edited into the /etc/fstab file.

Some of the more popular options, detailed below, are: `rsize`, `wsize`, `hard`, `soft`, and `timeo`.

rsize	Specifies the read buffer size. (Default is 1024.) Although there is an upper limit, the bigger the value, the faster the transfer speed.
wsize	Specifies the write buffer size. (Default 1024.)
hard	Even when the server is down, connection requests can continue. When the server is down, the message "server not responding" appears on the console.
soft	When there is no NFS server response for a while, the kernel is allowed to time out.
timeo	Specifies the length of the time out when soft has been set.

An example of /etc/fstab that uses the above options is:

```
nfssvr:/usr /mnt/usr nfs
    noauto,rw,rsize=8192,wsize=8192,soft,timeo=1000
```

In this case, the read and write buffer sizes are set to 8192 bytes each and after the server has been down for 1000 (ms) time out begins.

NFS Security

NFS is a kind of server that has elements problematic for system security. It is recommended that it be used only on a local network behind a firewall, and that the administrator carefully scrutinize the access permissions for /etc/exports.

In applying portmap access control, have a policy in place only to accept RPC calls from specifically designated clients.

References for NFS Server

- Sun Microsystems, http://www.sun.com/
- NFS has the five man pages listed below:

```
nfs             nfsd            fs
fstab           exports
```

Network Information Service (NIS)

NIS is an abbreviation for Network Information Service. It is used to share information about computers on the network, such as login names, passwords, home directories (/etc/passwd) and groups (/etc/group). This section discusses how to configure an NIS server and clients.

Name	**ypserv**
daemon	/usr/sbin/ypserv
configuration file	/etc/ypserv.conf
startup script	/etc/rc.d/init.d/ypserv
	/etc/rc.d/init.d/yppasswdd

Running portmap

To use NIS, you must be running portmap. Normally, when Turbolinux Server 6 for zSeries and S/390 boots, portmap is not started.

Enter the following command to do a portmap run check:

```
# /etc/rc.d/init.d/portmap status
```

If portmap is running, you will see:

```
portmap (pid 168) is running...
```

The portmap included in Turbolinux Server 6 for zSeries and S/390 references the TCP_Wrapper access control files, /etc/hosts.allow and /etc/host.deny, therefore, you must add the following line to the /etc/hosts.allow file:

```
portmap : 000.168.1.0/255.255.255.0 : allow
```

NIS Package Contents

To view the contents of the NIS (ypserv) package, use the `rpm -ql` option as shown below:

```
# rpm -ql ypserv
```

This command returns the listing shown below:

```
/etc/rc.d/init.d/yppasswdd
/etc/rc.d/init.d/ypserv
/etc/ypserv.conf
/usr/include/rpcsvc/ypxfrd.x
/usr/lib/yp
/usr/lib/yp/create_printcap
/usr/lib/yp/makedbm
/usr/lib/yp/match_printcap
/usr/lib/yp/mknetid
/usr/lib/yp/pwupdate
/usr/lib/yp/revnetgroup
/usr/lib/yp/yphelper
/usr/lib/yp/ypinit
/usr/lib/yp/ypxfr
/usr/lib/yp/ypxfr_1perday
/usr/lib/yp/ypxfr_1perhour
/usr/lib/yp/ypxfr_2perday
/usr/sbin/rpc_yppasswdd
/usr/sbin/yp/rpc_ypxfrd
/usr/sbin/yppush
/var/yp
/var/yp/Makefile
/var/yp/securenets
```

NOTE The contents of /usr/share/doc/packages/ypserv-1.3.9 are omitted. The relevant contents of /usr/share/man are found under *References for NIS* on page 2–63.

NIS Operation Confirmation

The operation of ypserv can be checked using the ps command as shown below:

```
ps aux | grep ypserv
```

If ypserv is operating properly, the above command returns the message shown below. Check that the program name ypserv appears at the far right of the listing.

```
root 3183 0.0 1.3 1140 404 tty1 S 22:47 0:00 grep ypserv
```

If you do not see something similar to the message shown above, either ypserv is not operating properly or it was not installed. Restart or reinstall it.

For more information, see *Managing System Processes* on page 1–25 and *Listing and Installing Packages* on page 1–33.

NIS Domain Settings

You can use the domainname command or edit /etc/sysconfig/network to configure NIS.

Run:

```
# domainname [domainname]
```

Or add this to the /etc/sysconfig/network file:

```
NISDOMAIN = [domainname]
```

NOTE The DNS domain name should be set to one different than the NIS domain name.

Server Settings

Check to see if /etc/ypserv.conf is available.

Next create the /var/yp/securenets file. Specify the usable scope of the NIS network by setting a combination of the netmask and the network address. Add the following line:

```
[netmask]    [network address]
```

The netmask that corresponds to the network address is specified in the netmask field, and the network address is specified in the

network address field. When the netmask is all 1s (that is, ones), the network address becomes the host address.

For example, on a private network 000.168.1.0/24, for NIS to work, the var/yp/securenets file would look like this:

```
255.255.255.255  127.0.0.1
255.255.255.0    000.168.1.0
```

In keeping with this objective, the /var/yp/Makefile will change. Start ypserv and run the NIS initialization dialog program.

```
# ypserv
# /use/lib/yp/ypinit -m
```

To run the server enter the following commands:

```
# /etc/rc.d/init.d/ypserv start
# /etc/rc.d/init.d/yppasswd start
```

To run the server the next time the system reboots, enable the server by using the chkconfig command.

```
# chkconfig --add ypserv
# chkconfig --add yppasswd
```

Client Settings

Add the following entry to the end of the /etc/passwd file.

```
+::::::
```

Add the following entry to the end of the /etc/group file.

```
+:::
```

Start ypbind using the following command:

```
# /etc/rc.d/init.d/ypbind start
```

To run ypbind on the next reboot, enable ypbind by using the chkconfig command as follows:

```
# chkconfig --add ypbind
```

Test to see if NIS is running properly like this:

```
# ypwhich
```

This command should return the following:

```
nissvr.turbolinux.com         <-- The NIS server name is
    displayed.
# ypcat passwd
user1:ylkXjOSM2R5rQ:501:501::/home/usr1:/bin/bash
user2:aqFAzdBEx8iZE:502:502::/home/user2:/bin/bash
```

How to Specify a Server by Means of ypbind

Edit the /etc/yp.conf file:

```
domain [domain name]   server [server name]
```

How to Add a New User to the NIS Server

A new user may be added to the NIS server with the useradd command:

```
# useradd [user name]
```

To refresh the NIS database, run:

```
# /use/lib/yp/ypinit -m
```

References for NIS

The following man pages are available for ypserv:

netgroup	ypserv.conf	makedbm	mknetid
pwupdate	revnetgroup	rpc.yppasswdd	rpc.ypxfrd
ypinit	yppasswdd	yppush	ypserv
ypxfr	ypxfrd		

OS Integration (Samba)

Samba is free, open source, software that can be used to integrate the Windows OS with Unix. Samba makes it possible to share the resources of Unix computers, such as Linux, with Windows so that an economical file server or print server, etc., can be set up for both systems. Also, by running a Samba client, the Windows resources can accessed from the UNIX side.

We will not discuss the Samba client here. For details, see the online manual found under /usr/share/doc/packages/samba-2.0.6/docs/faq.

Samba was developed by the Australian programmer, Andrew Tridgell, and first made public in 1992. The current development source is the Samba Team.

The version Turbolinux Server 6 for zSeries and S/390 uses is samba-2.0.7.

Samba Organization

Samba's main organization is as follows:

Daemon	Configuration File	Configuration
/usr/sbin/smbd	/etc/smbd.conf	File and printer sharing via the SMB protocol
/usr/sbin/nmbd	NetBIOS	WINS server functionality in NetBIOS

Samba startup mode

Samba may be run in standalone mode in which the Samba daemon is always on call or in superserver (inetd) mode. When starting Samba from the superserver, better security can be achieved by using TCP_Wrapper. These three modes are detailed below.

Standalone Mode

Turbolinux Server 6 for zSeries and S/390 starts Samba in standalone mode by default. In this manual our focus will be on standalone mode, but following are some hints for using Samba in Superserver mode.

Superserver Mode

To start Samba in superserver mode, add the following two lines to the /etc/inetd.conf file, then restart inetd.

```
netbios-ssn stream tcp  nowait root  /usr/sbin/smbd
netbios-sn  dgram  udp  wait   root  /usr/sbin/nmbd
```

Superserver Mode (with TCP_Wrapper)

To start Samba in superserver mode using TCP_Wrapper, add the following two lines to the /etc/inetd.conf file, then restart inetd.

```
swat-ssn stream tcp nowait root /usr/sbin/tcpd /usr/sbin/smbd
swat-sn  dgram  udp wait   root /usr/sbin/tcpd /usr/sbin/smbd
```

Starting and Stopping Samba

To utilize Samba, first enable SWAT in /etc/inetd.conf file

```
#vi /etc/inetd.conf
```

Then uncomment the line that contains the following:

```
swat   stream   tcp    nowait.400    root     /usr/sbin/swat  swat
```

After any changes are made in the inetd.conf file, one must restart the daemon. Use the following commands:

```
#killall -HUP /etc/inetd.conf
#/etc/rc.d/init.d/inetd.conf restart
```

Samba Startup Script (/etc/rc.d/init.d/smb)

The Samba startup script is /etc/rc.d/init.d/smb.

The startup script can take the following options: `start`, `stop`, `restart`, and `status`.

Whenever the Samba configuration is changed, you must restart Samba in order for the changes to go into effect.

To start Samba:

```
# /etc/rc.d/init.d/smb start
```

To stop Samba:

```
# /etc/rc.d/init.d/smb stop
```

To restart Samba:

```
# /etc/rc.d/init.d/smb restart
```

To check Samba's status:

```
# /etc/rc.d/init.d/smb status
```

Samba Operation Confirmation

Check if Samba is running with this command:

```
# ps aux | grep smb
```

The return should resemble:

```
root 766 0.2 0.2 1764 732 ? S 08:13 0:00 smbd -D
```

If you do not see a result like this, Samba may not be running or was not installed. Restart or install it.

For more information, see *Managing System Processes* on page 1–25 and *Listing and Installing Packages* on page 1–33.

Samba Configuration

In this section, we explain the settings on the server side, how to directly configure the configuration file.

Samba gets its settings from the /etc/smb.conf file, a plain text file composed of several sections, each of which is proceeded by a name surrounded by square brackets ([ ]). Its name becomes the unit for

file sharing. Each section which allows file sharing must correspond to a directory (except printers).

```
[public]
path=home/samba/public
```

This means that the settings in the [public] section apply to the /home/samba/public directory. Some section names are already reserved, namely, [global], [homes], and [printers].

[global]	Settings that apply to Samba as a whole
[homes]	Settings that apply to shared home directories
[printers]	Settings that apply to shared printers

When Samba is installed, the Samba configuration file, smb.conf is created under /etc. As given, smb.conf simply runs smbd and most of Samba's features are available to be used. But out consideration for security, one should edit smb.conf to reflect the environment in which it is used.

Standard /etc/smb.conf contains (abbreviated):

```
[global]
coding system = euc
client code page = 932
workgroup = WORKGROUP
server string = Samba %v
encrypt passwords = Yes
map to guest = Bad User
dns proxy = No
guest account = smbguest
[homes]
comment = %U's home directory
read only = No
browseable = No
[printers]
comment = All Printers
path = /var/spool/samba
print ok = Yes
```

```
browseable = No
[private]
comments = Private space; one can write one's own files.
path = /home/samba/private
read only = No
[public]
comment = Public space; anyone car write any files
path = /home/samba/public
guest ok = Yes
read only = No
force group = public
force create mode = 0664
force directory mode = 0775
[tmp]
comment = Read only file space
path = /tmp
guest ok = Yes
```

Here is an explanation section by section.

[global] Section

The [global] section appears at the top of smb.conf and lists settings that apply to Samba as a whole. We will explain the items that occur above:

- coding system

This determines how incoming Shift-JIS file names from Windows clients are converted into kanji encoding. Initially, it is set to euc. It is effective only if client code page is set to 932. You should leave this set to its initial default value.

- workgroup

This determines what Windows network NT domain name or workgroup name to which the Samba server belongs. The initial value is "WORKGROUP".

- server string

Some explanation about the server is given. When browsed by a Windows client on a networked computer, this string appears in the

"comment" column. The default is "Samba %v," where "%v" is replaced with the Samba version number. For Turbolinux Server 6 for zSeries and S/390, %v is Samba 2.0.7.

- encrypt passwords

This controls whether encrypted passwords will be negotiated when the client accesses the Samba server. This parameter can be set to either "Yes" or "No". The initial value is: "Yes".

Windows NT 4.0 (Service Packs 3 and above) and Windows 98 expect encrypted passwords by default, so this parameter must be set to "Yes". More about configuring for encrypted passwords will be given later.

- map to guest

This parameter tells smbd what to do with a request from a user that does not completely match a registered Unix user account. It can take any one of the three values listed below. The initial value is set to "Bad User".

Never	Login requests with an invalid password are rejected.
Bad User	Logins with an invalid password are rejected, unless the username does not exist, in which case it is treated as a guest login and mapped into the "guest account" (explained later).
Bad Password	Logins with an invalid password are treated as a guest login and mapped into the "guest account" (explained later).

ATTENTION	You should set this to either "Never" or "Bad User". Beware of using "Bad Password", because a user who enters an incorrect password will be logged in as "guest". No error message is displayed, so the user will wonder why he can no longer access his own files and directories.

- dns proxy

When a NetBIOS name cannot be found, this specifies whether to treat the name as the given DNS name or not. This parameter must be either "Yes" or "No". The default is "No".

- guest account

If the "guest ok=Yes" parameter is set in a section, this is the username that will be used for access to services. The default is "smbguest".

[homes] Section

This `[homes]` section holds settings that apply to user home directories.

- comment

Some explanation about the directory is given. When this directory is browsed by a Windows client on a networked computer, this string appears in the "comment" column. The default is "%U's Home directory", where the variable "%U" is replaced with the login name on the Samba server. If the user's name is 'jane', 'Jane's Home directory' is displayed.

- read only

Determines whether the home directory is read only or writing is also allowed. This parameter must be either "Yes" or "No". The default is "No".

- browseable

When this directory is browsed by a Windows client on a networked computer, this string appears in the "comment" column. The parameter must be either "Yes" or "No". The default is "No".

[printers] Section

This section holds settings that apply to the printers shares.

- comment

Some explanation about printers is given. When this directory is browsed by a Windows client on a networked computer, this string appears in the "comment" column. The default is "All Printers".

- path

Specifies the full path to the print spool directory. The default value is /var/spool/samba.

- print ok

Determines whether the client can write to the print spool file or not. This parameter must be either "Yes" or "No". The default is "Yes".

- browseable

When this directory is browsed by a Windows client on a networked computer, determines whether the printer will be displayed or not. This parameter must be either "Yes" or "No". The default is "No".

[private] Section

This section contains a sample of what may be used when you want to share private directories. It is not essential, but there may be occasions when it is needed.

The section name will be the name of shared directory. Here the shared directory is set to public.

- comment

This contains some explanation of the directory. When this directory is browsed by a Windows client on a networked computer, this string will be displayed in the "comment" column. The default is "Private space; one can write one's own files."

- path

Specifies the full path of the shared directories. The default is /home/samba/private.

- read only

Determines whether the home directory is read only or writing is also allowed. This parameter must be either "Yes" or "No". The default is "No".

[public] Section

This is an example of what may be used when you want to share public directories. It is not essential, but there may occasions when it is needed.

- comment

This contains some explanation of the directory. When this directory is browsed by a Windows client on a networked computer, this string will be displayed in a 'comment' box. The default is "Public space; anyone can write any files."

- path

Specifies the full path of the shared directories. The default is /home/samba/public.

- guest ok

Determines whether the access from a quest account is allowed or not. If it is allowed, access will be allowed from the user account set in the "guest account" parameter of the `[Global]` section.

- read only

Determines whether the home directory is read only or writing is also allowed. This parameter must be either "Yes" or "No". The default is "No".

- force group

Here you can force the setting of group ownership of the files and directories created in this directory. The default is "public".

- force create mode

Here you can force the setting of permissions of files created in this directory. The default is "644".

- force directory mode

Here you can force the setting of permissions of directories created in this directory. The default is "755."

[tmp] Section

This section is an example of the configuration you can use when you want to share a temporary directory (read only). It is not essential, but there may occasions when it is needed.

- comment

This contains some explanation of the directory. When this directory is browsed by a Windows client on a networked computer, this string will be displayed in a "comment" column. The default is "Read only file space."

- path

Specifies the full path of the shared directories. The default is /tmp.

- guest ok

Determines whether access from a quest account is allowed or not. If it is allowed, access will be allowed from the user account set in the "guest account" parameter of the *[global] Section*.

Encrypted Passwords

From Windows NT4.0 (Service Packs 3 and above), and in Windows98, encrypted passwords have been used as the default instead of the plain text passwords that used to be common on networks. Because these encrypted passwords were not compatible with Unix systems, access as usual was not possible. So it was necessary to make a password file especially for the Samba server. To enable encrypted passwords, as explained in a previous item, it is necessary to set the parameter "encrypt passwords=Yes" in the *[global] Section*.

To use encrypted passwords, there is a password file especially created for the Samba server called smbpasswd in which users can be registered. Some words of explanation follow.

NOTE To carry out these operations, you must be logged in as the superuser.

- Creating the smbpasswd file

Create an empty file called smbpasswd under /etc.

```
# touch /etc/smbpasswd
# chmod 600 /etc/smbpasswd
```

- User registration

In order for a user to register in the smbpasswd file, he or she must be registered on the system. Here we use the user ‘jane’ as an example.

```
# useradd jane
# smbpasswd -a jane

New SMB Password :
Repeat New SMB Password :
Added user jane
User jane enabled
```

To write a registered user into the smbpasswd file, run this command:

```
# cat /etc/passwd | mksbpasswd.sh > /etc/smbpasswd
```

After appending to the smbpasswd file, change the user's password.

```
#smbpasswd -a -e jane
```

Once changed, restart Samba.

File and Printer Sharing

There are four ways to set the format for file and printer sharing. You must decide which of the four values below to set for the "security" parameter in the [global] section. The "security" parameter is probably the most important setting in smb.conf.

- share

Shares are set up for directories and devices. They are of the same level as share settings in Windows95. Users not registered with the Samba server still have access. From a security standpoint, this is the most lenient configuration.

- user

Shares are set up user by user. Recognition is done by Samba, so a user who would gain access must be registered in advance with the Samba server. Furthermore, for access from Windows98 and NT SP3 and higher, encrypted passwords are needed. For details see *Encrypted Passwords* on page 2–74.

- server

This is the same configuration as for "user" above, but the recognition is done not by the Samba server but by another WindowsNT server. The user who would gain access must be already registered with another WindowsNT server. In this case, because the Samba server is seen as one client for the WindowsNT server, you need to make sure you have an additional WindowsNT client license.

- domain

This is the same configuration as for "user" above, but the Samba server is designated as a member of an already-existing WindowsNT domain. In short, the recognition is not done by the Samba server, but by the WindowsNT server (domain controller) instead. A user who would gain access must already be registered with this domain. In this case, because the Samba server is seen as one client for the WindowsNT server, you need to make sure you have an additional WindowsNT client license.

As in the previous item, when the "security" parameter is not made explicit, "`security=user`" is assumed.

Testing the settings

To check if the configuration file has been properly composed, run `testparm` as follows:

```
# testparm
```

This command returns the following:

```
Load smb config files from /etc/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[private]"
Processing section "[public]"
Processing section "[tmp]"
Loaded services file OK.

Press enter to see a dump of your service definitions
```

The display pauses at this point, so you can press ‘Enter’ to see the results of the test.

References for Samba

- The website for the Samba Team is located at http://www.samba.org/
- Several FAQ documents are available under /usr/share/doc/packages/samba-2.0.6/docs/faq
- The following man pages are available for Samba:

nmblookup	smbclient	smbrun	smbstatus
smbtar	testparm	testprns	smb.conf
samba	nmbd	smbd	smbpasswd
swat			

Printing Facilities (LPRng)

This section introduces the LPRng, print filters, printing facilities and tools included in Turbolinux.

On the Turbolinux Server 6 for zSeries and S/390, the traditional Berkeley LPR/LPD has been replaced with LPRng. Two newer printing systems have emerged - LPRng and CUPS.

LPRng is Berkeley LPR/LPD Next Generation, CUPS is the Common Unix Printing System. Turbolinux chose to use LPRng in preference to CUPS on the basis of maturity and for maximum backwards compatibility.

LPRng Package Contents

To view the contents of the LPRng package use the `rpm -ql` option as shown below.

```
# rpm -ql LPRng
```

The command results in the listing shown below:

```
/etc/lpd.conf
/etc/lpd.perms
/etc/rc.d/init.d/lpd
/usr/bin/cancel
/usr/bin/lp
/usr/bin/lpq
/usr/bin/lpr
/usr/bin/lprm
/usr/bin/lpstat
/usr/libexec/filters/lpbanner
/usr/libexec/filters/lpf
/usr/libexec/filters/pclbanner
/usr/libexec/filters/psbanner
/usr/sbin/checkpc
/usr/sbin/lpc|
/usr/sbin/lpd
/usr/sbin/lpraccnt
```

NOTE The contents of /usr/share/doc/packages/LPRng-3.6.26 are omitted from this listing. The relevant contents of /usr/man are listed under *References for LPRng* on page 2–81.

LPRng Startup Script

The startup script for LRPng is /etc/rc.d/init.d/lpd. View the contents of the default startup script using the less command as shown below:

```
# less /etc/rc.d/init.d/lpd
```

Start lpd by entering the following command:

```
# /etc/rc.d/init.d/lpd start
```

Stop lpd by entering the following command:

```
# /etc/rc.d/init.d/lpd stop
```

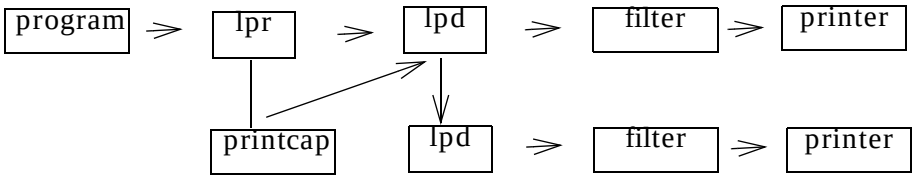
LPD Configuration File (/etc/lpd.conf)

At startup LPRng reads configuration information from a file which by default is the configuration file /etc/lpd.conf. Each line in the configuration file is commented out, that is, preceded by the hash (or pound) symbol [#] at the beginning of the line. To modify the default settings, delete the hash symbol and make the desired changes.

Print Spooling Overview

The following figure shows the flow of data between the individual components of the LPRng print spooling system. A program (or user) will use the lpr program to send a file to the lpd server over a TCP/IP connection.

Refer to the HTML documentation on LPRng on page 2–81 for comprehensive information on how to use printer filters with LPRng.



Sample Printcap Entry

Here is a sample printcap:

```
lp:lp=psqueue@printserver.acme.com
```

The printcap information tells the client programs that when a client wants to print a job on the lp printer, that these jobs should be sent to the psqueue on host printerserver.acme.com.

On the printerserver, the following printcap entry is used by the lpd server to do the printing.

```
psqueue:server
:lp=/dev/lp0
:sd=/var/spool/lpd/psqueue
:if=/usr/lib/filters/ifhp
```

Simple Server Printcap Example

```
#Local ASCII printer
lp1|printer
:server
:cm=Dumb printer
:lp=/dev/lp1
:sd=/var/spool/lpd/lp1
:if=log:af=acct
:if=/usr/lib/filters/lpf
:mx=0:sh:sf
```

LPRng checkpc Utility

The checkpc (check printcap file) is one of the most useful utilities in the LPRng package. It performs the following functions:

- Reads all the configuration and printcap files
- Tests whether devices are set up correctly.
- Sets the permissions for spool directories and device files
- Truncates the accounting and log files to a maximum size.
- Remove old entries from queue directories.

For a new installation, you will want to run

```
# checkpc -f -V
```

to set permissions right. The `-f` flag instructs the program to correct file permissions. If you do not run this as root, you will receive a warning about that fact and any chown(2) calls will (most likely) fail.

The program reports everything it changes. Since it is not too clever about some things (visit the man page), you should keep an eye on the output, and run it again if needed. If it keeps failing, change the permissions yourself.

```
# lpr -Pqueue@remote.dom.org <filename>
```

NOTE You must have lpd running. Lpd is the line printer daemon and is normally invoked at boot time from the rc(8) file.

References for LPRng

- http://www.astart.com/lprng/LPRng.html/
- http://www.astart.com/lprng/LPRng-HOWTO.html/
- man lpd
- man lpd.perms

Programming Languages (perl, Python)

This section addresses two programming languages that are included with Turbolinux Server 6 for zSeries and S/390.

Perl

Perl is a high-level programming language that derives primarily from the C programming language, and to a lesser extent from sed, awk, the Unix shell, and several other tools and languages.

Perl's process, file, and text manipulation facilities make it particularly well-suited for tasks involving quick prototyping, system utilities, software tools, system management tasks, database access, graphical programming, networking, and world wide web programming.

Perl Package Contents

To view the contents of the `Perl` package use the `rpm -ql` option as shown below:

```
# rpm -ql perl
```

The command returns the listing shown below:

```
/usr/bin/a2p
/usr/bin/c2ph
/usr/bin/find2perl
/usr/bin/h2ph
/usr/bin/h2xs
/usr/bin/perl
/usr/bin/perl15
/usr/bin/perl15.00503
/usr/bin/perldoc
/usr/bin/pl2pm
/usr/bin/pod2html
/usr/bin/pod2latex
/usr/bin/pod2man
/usr/bin/pstruct
/usr/bin/s2p
/usr/bin/sper15.00503
/usr/bin/suidperl
/usr/lib/perl5
/usr/lib/perl5/5.00503
/usr/lib/perl5/man
/usr/lib/perl5/man/man3
/usr/lib/perl5/site_perl
/usr/lib/perl5/site_perl/5.005
/usr/lib/perl5/site_perl/5.005/i386-Linux
```

NOTE The contents of /usr/lib/perl15/5.00503, /usr/lib/perl5/man/man3, and/usr/man/man1 are omitted. The man pages are listed below under *Perl References*.

Perl References

- The Perl Mongers' Perl Advocacy site, http://www.perl.org/
- O'Reilly Publications' Perl website, http://www.perl.com/
- Perl contains the man pages listed below:

a2p	perldsc	perlform	perllol	perltie
perl	perlembed	perlopentug	perlpod	perltoc
perlapio	perlfaq	perlguts	perlport	perltoot
perlbook	perlfaq1	perlhist	perlre	perltrop
perlbot	perlfaq2	perlreftut	perlref	perlvar
perlbug	perlfaq3	perllocale	perlipc	perlxs
perlcall	perlfaq4	perlthrtut	perlrun	perlmod
perldata	perlfaq5	perlxstut	perlsec	s2p
perldebug	perlfaq6	perlmodinstall	perlstyle	
perldelta	perlfaq7	perlmodlib	perlsub	
perldiag	perlfaq8	perlobj	perlsyn	
perldoc	perlfaq9	perlop	perlfunc	

There are approximately 150 man pages listed under /usr/lib/perl5/man. Use the `dir` command with the `more` option as follows to view the list:

```
# /usr/lib/perl5/man/man3 dir| more
```

Python

Python is an interpreted, iterative object-oriented programming language often compared to Tcl, Perl, Scheme and Java.

Python Package Contents

To view the contents of the Python package use the rpm -ql option as shown below.

```
# rpm -ql python
```

The command returns the listing shown below.

```
/usr/bin/python
/usr/bin/python1.5
/usr/lib/python1.5
/usr/lib/python1.5/lib-dynload <contents omitted>
/usr/lib/python1.5/lib-stdwin <contents omitted>
/usr/lib/python1.5/plat-linux-i386 <contents omitted>
/usr/lib/python1.5/
```

NOTE The contents of /usr/lib/python1.5, that are not themselves directories, are omitted.

Python Operation Confirmation

Python's operation can be checked using the ps command as shown below:

```
# ps aux | grep python
```

If Python is operating properly the above command should return something similar to the following message:

```
root 3926 0.0 1.3 1140 404 tty1 s 13:00 0:00 grep python
```

If you do not see this message, either Python is not operating properly or it was not installed. Restart or reinstall it. For more information, see *Managing System Processes* on page 1–25 and *Listing and Installing Packages* on page 1–33.

References for Python

- A major Python website is located at:
  http://www.python.org

There are no man pages for Python

Proxy/Caching Server (Squid)

Squid is a proxy caching server for web clients, supporting FTP, gopher, and HTTP data objects. Squid keeps metadata and especially hot objects cached in RAM, caches DNS lookups, supports non-blocking DNS lookups, and implements negative caching of failed requests.

Squid supports SSL, extensive access controls, and full request logging. By using the lightweight Internet Cache Protocol, Squid caches can be hierarchically linked to other Squid-based proxy servers for streamlined caching of pages.

Squid consists of a main server program, squid; a Domain Name System (DNS) lookup program, dnsserver; some optional programs for rewriting requests and performing authentication; and some management and client tools.

When Squid starts up, it spawns a configurable number of dnsserver processes, each of which can perform a single blocking DNS lookup. This reduces the amount of time the cache waits for DNS lookups.

Squid Package Contents

To view the contents of the Squid package use the `rpm -ql` option as shown below.

```
# rpm -ql squid
```

The command results in the listing shown below. (Contents of the errors, icons, and /usr/share/doc/packages/squid directories are omitted.)

```
/etc/logrotate.d/squid
/etc/rc.d/init.d/squid
/etc/squid/errors
/etc/squid/mib.txt
/etc/squid/mime.conf
/etc/squid/mime.conf.default
/etc/squid/squid.conf
/etc/squid/squid.conf.default
/usr/lib/squid/cachemgr.cgi
/usr/lib/squid/dnsserver
/usr/lib/squid/errors
/usr/lib/squid/icons
/usr/lib/squid/unlinkd
/usr/sbin/client
/usr/sbin/squid
/usr/share/doc/packages
/var/log/squid
/var/spool/squid
```

Squid Configuration

The Squid control files are located under /etc/squid. The main configuration control file is called squid.conf and will nearly always require customization. The most commonly customized settings include access controls.

To view a chapter on configuration basics, visit the *Squid User's Guide* at http://www.squid-cache.org/.

Default Settings (/etc/squid/squid.conf)

The default setup configuration is located at /etc/squid/squid.conf on the Turbolinux Server. To view it, enter the following command:

```
# less /etc/squid/squid.conf
```

This is a selection of the results. The Squid configuration page contains a great deal of useful internal documentation.

```
#http_port              3128
#icp_port               3130
#htcp_port              4827
#mcast_groups           239.128.16.128
#cache_peer             hostname        type    http_port       icp_port        options
icp_query_timeout       (ms) (defauly=0)
```

Access Controls. You will want to add access controls for your local network to enable your users to use the Squid proxy cache. For example, if your local network uses address 000.168.1.0 with a netmask of 255.255.255.0, you would add the following:

```
acl mynet src 000.168.1.0/255.255.255.0
http_access allow mynet
```

Neighbor Selection Algorithm. Please read the documentation in /etc/squid/squid.conf regarding use of this parameter.

Squid Operation Confirmation

Squid's operation can be checked using the ps command as shown below:

```
# ps aux | grep squid
```

If Squid is operating properly the above command should return something similar to the following message:

```
root 3926 0.0 1.3 1140 404 tty1 s 13:00 0:00 grep squid
```

If you do not see this message, either Squid is not operating properly or it was not installed. Restart or reinstall it. For more information, see *Managing System Processes* on page 1–25 and *Listing and Installing Packages* on page 1–33.

Squid References

- Squid Web Proxy Cache home page, http://www.squid-cache.org/
- Squid User's Guide, http://ww.squid-docs.soureforge.net/latest/html/book1.htm/
- A Squid FAQ is available at http://squid.nlanr.net/Doc/FAQ/FAQ-1.html/
- There are no `man` pages in the Squid package.

RPC Program Number Converter (Portmap)

The portmap server converts RPC program numbers to DARPA protocol numbers. It is necessary to run it in order to use a server, such as NFS and NIS, that employs RPC calls.

Portmap Package Contents

To view the contents of the portmap package use the `rpm -ql` option as shown below:

```
# rpm -ql portmap
```

The command returns the listing shown below.

```
/etc/rc.d/init.d/portmap
/etc/rc.d/rc0.d/K89portmap
/etc/rc.d/rc1.d/K89portmap
/etc/rc.d/rc2.d/K89portmap
/etc/rc.d/rc3.d/S11portmap
/etc/rc.d/rc5.d/S11portmap
/etc/rc.d/rc6.d/K89portmap
/sbin/portmap
/usr/man/man8/pmap_dump8
/usr/man/man8/pmap_set.8
/usr/man/man8/portmap.8
/usr/sbin/pmap_dump
/usr/sbin/pmap_set
/usr/share/doc/packages/portmap-4.0
/usr/share/doc/packages/portmap-4.0/BLURB
/usr/share/doc/packages/portmap-4.0/CHANGES
/usr/share/doc/packages/portmap-4.0/README
/usr/share/doc/packages/portmap-4.0
```

Portmap Startup Script and Options

The portmap startup script is located in /etc/rc.d/init.d/portmap. Use the following command set to access it:

```
# /etc/rc.d/init.d/portmap [start|stop|restart]
```

Once you change the settings of portmap, you must restart it in order for the changes to go into effect.

Access Control

Portmap is extremely convenient, but from the security standpoint it is also extremely problematic. In the default installation of portmap under Turbolinux Server 6 for zSeries and S/390, the TCP_Wrapper library can be used for access control.

Once running under the initial settings of Turbolinux Server 6 for zSeries and S/390, portmap does not accept requests from other clients. Because of this, you will not be able to use servers such as NFS and NIS by simply running portmap. You will need to modify /etc/hosts.allow, the access control file, for TCP_Wrapper by following the example below.

In this example, access is allowed from the `.turbolinux.com` domain or from host 000.168.0.1.

```
portmap: .turbolinux.com
portmap: 000.168.0.1
```

The changes in the access control file will be reflected without restarting portmap.

ATTENTION If portmap crashes, all rpc services must be restarted.

NOTE Portmap is not started from tcpd or from the superserver, but portmap itself can reference /etc/hosts.allow and /etc/hosts.deny.

Portmap References

In the man pages for portmap, one finds the options for the various commands and references to related (non-portmap) man pages.

- Portmap has three man pages: `portmap, pmap_set and pmap_dump`.

SQL Database Management System (PostgreSQL)

PostgreSQL is a sophisticated Object-Relational DBMS, supporting almost all SQL constructs, including subselects, transactions, and user-defined types and functions. It is the most advanced open-source database available anywhere.

PostgreSQL uses a client/server model of communication. That means that a PostgreSQL server continually runs, waiting for client requests. The server processes the request and returns the result to the client.

Because the PostgreSQL server runs as an independent process on the computer, there is no way for a user to interact with it directly. Instead, there are client applications designed specifically for user interaction. You can interact with PostgreSQL using the psql interface.

PostgreSQL is an enhancement of the POSTGRES database management system, a next-generation DBMS research prototype.

NOTE PostgreSQL's daemon is named postmaster.

PostrgreSQL Package Contents

To view the contents of the PostgreSQL package, use the `rpm -ql` option as shown below:

```
rpm -ql postgresql
```

A truncated version of this command's results are shown below.

```
/usr/bin/createdb
/usr/bin/createlang
/usr/bin/createuser
/usr/bin/destroydb
/usr/bin/destroylang
/usr/bin/destroyuser
/usr/bin/pg_dump
/usr/bin/pg_dumpall
/usr/bin/pg_id
/usr/bin/psql
```

PostgreSQL Startup Script (/etc/rc.d/init.d/postgresql/)

The startup script for postgresql is:

```
/etc/rc.d/init.d/postgresql {start|stop}
```

To view the default startup script, enter the following command:

```
#less /etc/rc.d/init.d/postgresql
```

The following is an abbreviated listing of the return:

```
./etc/rc.d/init.d/functions
    ./etc/sysconfig/network
    [ ${NETWORKING} = "no" ] && exit 0
    [ -f /usr/bin/postmaster ] || exit 0
    case "$1" in
          start
          echo -n "Checking postgresql installation: "
                if [ -f /var/lib/pgsql/PG_VERSION ] && [ -d /var/
    lib/pgsql/base/template 1 ]
          then
                if [ 'cat /var/lib/pgsql/PG_VERSION' != '6.5' ]
                then
                      echo "old version. Need to Upgrade."
```

```
                    echo "see /usr/doc/postgresql-6.5.2/
README.rpm for more information."
                    exit 1
              else
                    echo "looks good!"
              fi
       else
              echo "no database files found."
              if [ ! -d /var/lib/pgsql ]
              then
                    mkdir -p /var/lib/pgsql
       fi
       su -l postgres -c '/usr/bin/initdb --pglib=/usr/lib/
pgsql --pgdata=/var/lib/pgsql'
       fi
pid='pidof postmaster
if [ $pid ]
then
       echo "Postmaster already running."
else
       rm -f /tmp/.s.PGSQL.* > /dev/null
       echo -n "Starting postgresql service: "
       su -l postgres -c '/usr/bin/ postmaster -i -S -D/var/
lib/pgsql/'
       sleep 1
       pid= 'pidof postmaster'
       if [ $pid ]
       then
              echo "failed."
       fi
       ;;
stop)
       echo -n "Stopping postgresql service: "
       killproc postmaster
       sleep 2
       rm -f /var/run/postmaster.pid
       rm -f /var/lock/subsys/postgresql
       echo
       ;;
status)
       status postmaster
       ;;
```

```
restart)
      $0 stop
      $0 start
      ;;
*)
      echo "Usage: postgresql start|stop|status|restart)"
      exit 1
esac
exit 0
```

PostgreSQL Operation Confirmation

PostgreSQL's operation can be checked using the ps command as shown below.

```
ps aux | grep postgresql
```

If postgreSQL is operating properly the above command should return the message shown below. Check that the program name `postgresql` appears at the far right of the listing.

```
root  2890 0.0  1.3 1140 404 tty1 S  15:54 0:00 grep postgresql
```

If you do not see something similar to the message shown above, either postgreSQL is not operating properly or it was not installed. Restart or reinstall it. For more information, see *Managing System Processes* on page 1–25 and *Listing and Installing Packages* on page 1–33.

PostgreSQL Configuration Parameters

The following information is from the PostgreSQL Administrator's Guide found at http://postgresql.readysetnet.com/users-lounge/docs/6.5/admin/

The full set of parameters available in configure can be obtained by typing:

```
$ ./configure --help
```

The following parameters may be of interest to installers:

Directory and file names

`--prefix=PREFIX`	install architecture-independent files in PREFIX [/usr/local/pgsql]
`--bindir=DIR`	user executables in DIR [EPREFIX/bin]
`--libdir=DIR`	object code libraries in DIR [EPREFIX/lib]
`--includedir=DIR`	C header files in DIR [PREFIX/include]
`--mandir=DIR`	man documentation in DIR [PREFIX/man]

Features and packages

`--disable-FEATURE`	do not include FEATURE (same as --enable-FEATURE=no)
`--enable-FEATURE[=ARG]`	include FEATURE [ARG=yes]
`--with-PACKAGE[=ARG]`	use PACKAGE [ARG=yes]
`--without-PACKAGE`	do not use PACKAGE (same as --with-PACKAGE=no)

--enable and --with options recognized

`--with-template=template`	use operating system template file see template directory
`--with-includes=`*`incdir`*	site header files for tk/tcl, etc in DIR
`--with-libs=`*`incdir`*	also search for libraries in DIR

`--with-libraries=`*`libdir`*	also search for libraries in DIR
`--enable-locale`	enable locale support
`--enable-recode`	enable cyrillic recode support
`--with-mb=`*`encoding`*	enable multi-byte support
`--with-pgport=`*`portnum`*	change default startup port
`--with-maxbackends=`*`n`*	set default maximum number of server processes
`--with-tcl`	build Tcl interfaces and pgtclsh
`--with-tclconfig=`*`tcldir`*	tclConfig.sh and tkConfig.sh are in DIR
`--with-perl`	build Perl interface
`--with-odbc`	build ODBC driver package
`--with-odbcinst=`*`odbcdir`*	change default directory for odbcinst.ini
`--enable-cassert`	enable assertion checks (debugging)
`--with-CC=`*`compiler`*	use specific C compiler
`--with-CXX=`*`compiler`*	use specific C++ compiler
`--without-CXX`	prevent building C++ code

Some systems may have trouble building a specific feature of PostgreSQL. For example, systems with a damaged C++ compiler may need to specify --without-CXX to instruct the build procedure to skip construction of libpq++.

References for PostgreSQL

- PostgreSQL web home page, http://www.postgresql.org/
- PostgreSQL user's guide, http://postgresql.readysetnet.com/users-lounge/docs/6.5/admin/

- PostgreSQL has approximately 70 man pages listed in the table below:

close	create_table	createdb	delete_index	grant
cluster	create_table_as	createlang	delete_language	insert
commit	create_trigger	createuser	delete_operator	listen
copy	create_type	deletedb	delete_rule	load
create_aggregate	create_user	deletelang	delete_sequence	lock
create_database	create_version	deleteuser	delete_table	move
create_function	declare	pgdump	delete_trigger	notify
create_index	delete	psql	delete_type	reset
create_language	drop	abort	delete_user	revoke
create_operator	delete_aggregate	alter_table	end	rollback
create_rule	delete_database	alter_user	explain	select
create_sequence	delete_function	begin	fetch	select_into
create_view	delete_view	pgdumpall	set	show
update	vacuum	unlisten	postmaster	

SQL Database Server (MySQL)

MySQL is a multi-user, multi-threaded SQL database server. SQL (Structured Query Language) is the most popular and standardized database language in the world. MySQL is a client/server implementation that consists of a server daemon, mysqld, and several client programs and libraries.

SQL is a standardized language that makes it easy to store, update and access information. For example, you can use SQL to retrieve product information and store customer information for a web site. MySQL is also fast and flexible enough to allow you to store logs and pictures in it.

MySQL is speedy, robust, and easy to use. It was originally developed because there was a need for a SQL server that could handle very large databases an order of magnitude faster than what any database vendor could offer on inexpensive hardware.

MySQL is built on a set of routines that have been used in a highly demanding production environment for many years. Although MySQL is still under development, it offers a rich and highly useful function set.

The package containing MySQL is found on the *Turbolinux Server 6 for zSeries and S/390 Install CD* under /RPMS/s390/MySQL-3.22.32-5.s390.rpm.

Name	**MySQL**
daemon	/usr/sbin/mysqld
configuration file	See *MySQL Configuration* on page 2–103
startup script	/usr/bin/safe_mysqld

MySQL Package Contents

The contents of the MySQL package can be viewed using the `rpm -ql` option as shown below:

```
# rpm -ql MySQL
```

The command results in the listing shown below.

```
/etc/logrotate.d/mysql
/etc/rc.d/init.d/mysql
/usr/bin/insamchk
/usr/bin/isamlog
/usr/bin/mysql_fix_privilege_tables
/usr/bin/mysql_installdb
/usr/bin/mysql_setpermission
/usr/bin/mysql_zap
/usr/bin/mysqlbug
/usr/bin/perror
/usr/bin/replace
/usr/bin/resolveip
/usr/bin/safe_mysqld
/usr/sbin/mysqld
/usr/share/doc/packages/MySQL-3.22.32
/usr/share/info/mysql.info.gz
/usr/share/mysql
```

NOTE The contents of the /usr/share/doc/packages/MySQL-3.22.32 and /usr/share/mysql directories are omitted.

MySQL Startup Options

MySQL's startup script is located at /etc/rc.d/init.d/mysql.

Follow this command to execute the MySQL daemon:

```
# /etc/rc.d/init.d/mysql start
```

The following options can be specified when starting MySQL:

`-#|--debug=logfile`	Employ the specified debug log
`-T|--debug-info`	set debug mode
`-P|--port=pnum`	employ the specified port number for connecting to the database server.

Other MySQL Scripts

/usr/bin/safe_mysqld	executable shell script for starting mysqld safely
/usr/bin/mysqld	executable
/usr/sbin/mysqld	executable

MySQL Configuration

The following information is from the MySQL online documentation found at http://www.mysql.com/Manual_chapter/manual_toc.html/.

The configure script gives you a great deal of control over how you configure your MySQL distribution. Typically you do this using options on the configure command line. You can also affect configure using certain environment variables. For a list of options supported by configure, run this command:

```
shell> ./configure --help
```

Some of the more commonly-used configure options are described below.

To compile just the MySQL client libraries and client programs and not the server, use the `--without-server` option:

```
shell> ./configure --without-server
```

If you do not have a C++ compiler, MySQL will not compile (it is the one client program that requires C++). In this case, you can remove the code in configure that tests for the C++ compiler and then run `./configure` with the `--without-server` option. The compile step will still

try to build MySQL, but you can ignore any warnings about `mysql.cc'. (If `make` stops, try `make -k` to tell it to continue with the rest of the build even if errors occur.)

If you do not want your log files and database directories located under /usr/local/var, use one of the following configure commands:

```
shell> ./configure --prefix=/usr/local/mysql
     shell> ./configure --prefix=/usr/local
                --localstatedir=/usr/local/mysql/data
```

The first command changes the installation prefix so that everything is installed under /usr/local/mysql rather than the default of /usr/local.

The second command preserves the default installation prefix, but overrides the default location for database directories (normally /usr/local/var) and changes it to /usr/local/mysql/data.

If you are using Unix and you want the MySQL socket located somewhere other than the default location (normally in the directory /tmp or /var/run), use a configure command like this:

```
shell> ./configure --with-unix-socket-path=/usr/local/mysql/tmp/
    mysql.sock
```

NOTE The given file must be an absolute pathname!

If you want to compile statically linked programs (for example, to make a binary distribution or to get more speed), run configure like this:

```
shell> ./configure --with-client-ldflags=-all-static --with-
    mysqld-ldflags=-all-static
```

If you are using gcc and don't have libg++ or libstdc++ installed, you can tell configure to use gcc as your C++ compiler:

```
shell> CC=gcc CXX=gcc ./configure
```

When you use gcc as your C++ compiler, it will not attempt to link in libg++ or libstdc++. If the build fails and produces errors about your compiler or linker not being able to create the shared library libmysqlclient.so.# (where # is a version number), you can work around this problem by giving the --disable-shared option to configure. In this case, configure will not build a shared libmysqlclient.so.# library.

You can configure MySQL not to use default column values for non-NULL columns (that is, columns that are not allowed to be NULL). This causes INSERT statements to generate an error unless you explicitly specify values for all columns that require a non-NULL value. To suppress use of default values, run configure like this:

```
shell> CXXFLAGS=-DDONT_USE_DEFAULT_FIELDS ./configure
```

By default, MySQL uses the ISO-8859-1 (Latin1) character set. To change the default set, use the --with-charset option:

```
shell> ./configure --with-charset=CHARSET
```

CHARSET may be one of big5, cp1251, cp1257, czech, danish, dec8, dos, euc_kr, gb2312, gbk, german1, hebrew, hp8, hungarian, koi8_ru, koi8_ukr, latin1, latin2, sjis, swe7, tis620, ujis, usa7, or win1251ukr. If you want to convert characters between the server and the client, you should take a look at the SET OPTION CHARACTER SET command.

ATTENTION If you change character sets after having created any tables, you will have to run `myisamchk -r -q` on every table. Your indices may be sorted incorrectly otherwise. (This can happen if you install MySQL, create some tables, then reconfigure MySQL to use a different character set and reinstall it.)

To configure MySQL with debugging code, use the `--with-debug` option:

```
shell> ./configure --with-debug
```

This causes a safe memory allocator to be included that can find some errors and that provides output about what is happening.

If your client programs are using threads, you need to also compile a thread-safe version of the MySQL client library with the `--with-thread-safe-client` configure options. This will create a `libmysqlclient_r` library with which you should link your threaded applications.

For more information, see the MySQL Reference Manual and the other references listed below.

References for MySQL

- MySQL Reference Manual, http://www.mysql.com/Manual_chapter/manual_toc.html/
- Developer Shed's "Beginning MySQL Tutorial", http://www.devshed.com/Server_Side/MySQL/Intro/
- MySQL has man pages for the items listed below:

`mysqlaccess` `mysqladmin` `mysqlbug` `mysqld` `mysqldump` `mysqlshow` `mysql2mysql` `safe_mysqld`

Super Server (inetd)

Currently, inetd is one of the most popular super server programs. By default Turbolinux Server 6 for zSeries and S/390 installs inetd and sets it as the system program at bootup time.

When the system is initialized, inetd runs the start-up script (/etc/rc.d/init.d/inet). Inetd examines the port defined in the configuration file (/etc/inetd.conf) and waits for a connection request (request packet). When there is a connection request, inetd listens to the specified port and responds with a service by executing a server program. The server program then takes over the connection request itself while the service socket handles the normal input, output and error service requests. When the server program terminates, the inetd program returns to monitoring the port's state.

The main server programs, including `ftp` and `telnet`, are used for controlling and managing services. Turbolinux Server is a *super server* in that it is a high-level program acting as a server to other servers. The program includes inetd and its expanded function version, TCP_wrapper, name server, web server, samba, and NFS server. System managers should be sure they have a full understanding of the super server and these programs.

Server Program

Linux and Unix operate by executing commands from the user's computer via client programs.

Client programs send requests to the server. The server program receives client requests and responds to the client by executing super server programs to fulfill client requests. For example, in the case of FTP, the client program ProFTP calls the server program in.ftpd which executes the requested services.

TCP_Wrapper

Opinion is divided on whether TCP_Wrapper should be called a super server. The reasoning behind calling TCP_Wrapper a super server is:

- From the point-of-view of functionality, it offers various server features for the control and management of servers.
- In terms of connectivity, it permits close connections and program interchange with other super servers (`inetd`, etc.).

Super Server Mode and Standalone Mode

Super servers permit the direct control and management of server programs. In standalone mode the control and management commands are executed directly. On the other hand, super server mode applies to programs executed under the control of the super server. Whether a program is run in super server or standalone mode depends on its function and role.

Super Server Functions and Role

- System load reduction

When programs are executed in standalone mode, each server program must be resident in memory and this consumes a substantial amount of memory. Typically all the server programs are not running. In addition, immediately after requests are processed the system returns to the wait state while the memory remains occupied. The super server resides in memory as an agent on behalf of the server programs where it can manage memory usage and reduce the system load.

- System management load reduction

In cases when numerous server programs are executed in standalone mode each server program must be managed individually. It is important for system administrators to be able to respond to the growing requirement for services and the resulting proliferation of different server programs. A super server

gives the administrator a means of uniform control and management of these programs.

- System redundancy and stability

In super server mode, should a server program fail to execute properly, for whatever reason, upon the next request the service will be re-executed. This feature contributes to the system's redundancy and stability.

Many new services are being developed everyday and many more server programs are becoming necessary. These programs as well as non-executable programs can be managed in a unified environment by the super server which includes this ability as one of its basic features. In practice, the various implementations of super servers, such as Linux and FreeBSD, by default are set to operate server programs in the super server mode.

Super Server Issues

While the server program is executing, and/or ending, a program in response to requests, it is possible for the load to actually increase. This could happen under the following conditions:

- When persistent requests for a connection are made
- When the program runtime initialization process takes a long time

In cases when a particular service is putting a heavy load on the system, it is more efficient to run in standalone mode. A typical example is a web server which is quite likely to receive persistent requests for a particular connection. The mail server program `sendmail` is also frequently executed in standalone mode for the same reason. According to the situation the appropriate mode should be selected.

Name	**inetd**	
daemon	/usr/sbin/inetd	
configuration file	/etc/inetd.conf	
startup script	/etc/rc.d/init.d/inet	
log file	/var/lock/subsys/inet	
related files	/etc/services	service name and port number list
	/etc/protocols	protocol name and port number list

inetd Startup Options

The following options can be specified when starting inetd:

-d	set debug mode
-l	create a log file
-c maximum	specifics the maximum number of daemons at startup time. The maximum is specified for each service. The ax-child parameter can be used as a superscript.
-C rate	specify in 1 minute units the maximum IP address daemons startup default setting. The maximum-connections-per-ip-per-minute parameter allowed for each service can be used as a superscript
-R rate	specify in 1 minute units the maximum services at startup. Default is 256.
-a	specify the IP address to listen to.
-p	specifies the file holding the process IDs other than the default process.

inetd Configuration File (/etc/inetd.conf)

At startup inetd reads configuration information from a file which by default is the configuration file /etc/inetd.com. Configuration fields are separated by either a space or tab character. Comments are preceded by the hash (or pound) symbol [#] at the beginning of the line. Each line in the configuration file is listed below.

```
service_name  socket_type  protocol  flag  user  server_path
args
```

Below is an explanation of each field.

service name	Indicates the service name such as ftp, telnet, etc. The list of host and port names is recorded in /etc/services
socket_type	Indicates the type of socket such as stream (stream type), dgram (datagram type), etc.
protocol	Indicates the type of protocol used such as tcp, udp, etc selected from the list of protocols stored in /etc/protocols.
flag	A marker for either the wait or nowait flag; allows only to dgram (datagram) type sockets. The wait flag indicates a wait while a request/response exchange take places; the nowait flag indicates that a request will have no wait for a response.
user	Users such as root, nobody, etc. indicate the user's level of permission to access services.
server_path	The full path name of the server program executed.
args	The server program startup name and the options specified in the server_path.

Default Settings (/etc/inetd.conf)

Below is a specific example of a default setup configuration created by /etc/inetd.conf in Turbolinux Server.

Enter the following command:

```
# less /etc/inetd.conf
```

The following is an abbreviated listing of the result:

#dtalk	stream	tcp	nowait	nobody	/usr/sbin/tcpd	in.dtalkd
#exec	stream	tcp	nowait	root	/usr/sbin/tcpd	in.rexecd
#ftp	stream	tcp	nowait	root	/usr/sbin/tcpd	in.ftpd -1 -a
#gopher	stream	tcp	nowait	root	/usr/sbin/tcpd	gn
#imap	stream	tcp	nowait	root	/usr/sbin/tcpd	imapd
#login	stream	tcp	nowait	root	/usr/sbin/tcpd	in.rlogind
#nntp	stream	tcp	nowait	root	/usr/sbin/tcpd	in.nntpd
#ntalk	dgram	udp	wait	root	/usr/sbin/tcpd	in.ntalkd
#pop-3	stream	tcp	nowait	root	/usr/sbin/tcpd	ipop3d
#shell	stream	tcp	nowait	root	/usr/sbin/tcpd	in.rshd
#smtp	stream	tcp	nowait	root	/usr/bin/smtpd	smtpd
#talk	dgram	udp	wait	root	/usr/sbin/tcpd	in.talkd
#telnet	stream	tcp	nowait	root	/usr/sbin/tcpd	in.telnetd -h

With Turbolinux Server the defaults set by /etc/inetd.conf are set to the optimum configuration for a secure setup. There is no need to run all of the available daemons; each service is commented in a line preceded by a hash mark [#].

Enable Changes to Default inetd Settings (restart)

Changes and edits to inetd.conf do not automatically enable the new default settings. It is necessary to restart inetd to make the changes take effect. Below we shown several ways of restarting. By examining the startup script inetd, you can see that the methods use essentially the same process (`killall -HUP inetd`). It is possible to use any of the methods shown.

```
# killall -HUP inetd

# /etc/rc.d/init.d/inet restart

# /etc/rc.d/init.d/inet reload
```

Configuration Example

The example below shows the steps in setting up service permissions using ftp. It uses the case of a single client-host connection in which the server host IP address is 000.168.1.52 and the client IP address is 000.168.1.53 in which ftp provides the permissions.

1. Locate the place in /etc/inetd.conf where you find the line with ftp commented out (with # symbol at the beginning of the line) and remove the # symbol.

   ```
   #ftp  stream  tcp  nowait  root  /usr/sbin/tcpd  in.ftpd -
       l  -a
   ```

   With an editor program change the line as shown below and save the file.

   ```
   ftp  stream  tcp  nowait  root  /usr/sbin/tcpd  in.ftpd -l
       -a
   ```

2. To enable these changes input the command below to restart inetd.

   ```
   # killall  -HUP  inetd
   ```

3. Alter the TCP_Wrapper configuration file. At the end of the /etc/hosts.allow file append the following line:

   ALL: 000.168.1.52: allow

inetd Package Contents

The inetd component is not included in the main Turbolinux package. To view the contents of the inetd package use the `rpm -ql` option as shown below.

```
# rpm -ql inetd
```

The command results in the listing shown below.

```
/etc/rc.d/init.d/inet
/usr/sbin/inetd
/usr/share/doc/packages/inetd-0.16
/usr/share/doc/packages/inetd-0.16/BUGS.gz
/usr/share/doc/packages/inetd-0.16/ChangeLog.gz
/usr/share/doc/packages/inetd-0.16/README.gz
/usr/man/man5/inetd.conf.5.gz
/usr/man/man8/inetd.8.gz
```

inetd Startup Script

The startup script for inetd is /etc/rc.d/init.d/inet. The contents of the default startup script is shown below in slightly abbreviated form.

```
./etc/rc.d/init.d/functions
./etc/sysconfig/network
if [ ${NETWORKING} = “no” ]
then
exit O
fi
[ -f /usr/sbin/inetd ] || exit O
case “$1” in
start)
echo -n “Starting INET services; “
daemon inetd
echo
touch /var/lock/subsys/inet
;;
stop)
echo -n “Stopping INET services; “
killproc inetd
echo
rm -f /var/lock/subsys/inet
;;
```

```
status)
status inetd
;;
restart|reload)
killall -HUP inet
;;
*)
echo “Usage: inet|start|stop|status|restart|reload}”
exit 1
esac
exit 0
```

Startup Script Command Line Options

Command line options for the startup script /etc/rc.d.init.d/inet are shown below.

/etc/rc.d/init.d/inet start	starts inetd
/etc/rc.d/init.d/inet stop	stops inetd
/etc/rc.d/init.d/inet status	displays status of inetd
/etc/rc.d/init.d/inet restart	restarts inetd
/etc/rc.d/init.d/inet reload	reloads inetd

As you can see by examining the startup script, restart and reload perform the same operation.

inetd Operation Confirmation

In Turbolinux Server inetd is set to operate by default. Its operation can be checked using the ps command as shown below.

```
ps aux | grep inetd
```

If inetd is operating properly the above command should display the message shown below. Check that the program name inetd appears at the far right of the listing.

```
root  163  0.0  1.2  792 400 ?  S  15:20   0:00   inetd
```

If you do not see the message shown above either inetd is not operating properly or it was not installed. Restart inetd or reinstall it.

For more information, see *Managing System Processes* on page 1–25 and *Listing and Installing Packages* on page 1–33.

inetd References

- An inetd introduction is available at http://www.uwsg.iu.edu/usail/network/services/inetd.html/
- `man inetd`
- `man inetd.conf`

Index

A

access.conf file 2-40
Account
  deleting 1-6
  root 1-4
  switching 1-6
Apache 2-35

B

BIND 2-8
BIND Configration File 2-8
BIND Package Contents 2-7

C

cat (concatenate)
  command 1-17
cd
  command 1-11
Changing file names 1-15
Changing password 1-5
Changing user accounts 1-4
checkpc Utility 2-81
Command
  kill (stopping processes) 1-26
Commands
  cat (concatenate), less, more 1-17
  cd (change directory) 1-11
  cp (copy) 1-13
  df and du (disk usage) 1-29
  find 1-20
  grep (searching for strings) 1-21
  gzip (compressing and expanding files) 1-22
  kill (stopping processes) 1-26
  man (online manuals) 1-35
  mkdir (make directory) 1-16
  mv (move) 1-14, 1-15
  ps (processes) 1-25
  pwd (print working directory) 1-12
  rm and rmdir (remove files) 1-16
  rpm and turbopkg (packages) 1-33
  tar (creating and extracting files) 1-23
Compressing and expanding (gzip) files 1-22
cp (copy)
  command 1-13
Creating a directory 1-16
Creating and extracting (tar) files 1-23
customer support xi

D

daemons
  about 2-4
Deleting
  files 1-16
  user accounts 1-6
Devices 1-28
Directories
  deleting 1-16
  working with 1-10
Directory
  changing 1-11
  creating 1-16
Disk usage (df, du)
  command 1-29

F

Files
  changing names 1-15
  copying 1-13
  deleting 1-16
  moving 1-14
  viewing 1-17
  working with 1-10
find
  command 1-20

G

grep (searching for strings)
  command 1-21

gzip (compressing and expanding files)
command 1-22

H

Halt system
at login screen 1-9
Help
online manuals 1-35
httpd
configuration file 2-36
startup 2-35
httpd Startup Script 2-35

I

inetd 2-107
configuration file 2-111
defaults 2-89, 2-112
package contents 2-88, 2-114
install
RPM packages 2-3
inverse file (reverse) 2-18

K

kill (stopping processes)
command 1-26

L

less
command 1-17
Logging in 1-2
loop-back file 2-16
LPRng 2-78

M

mkdir (make directory)
command 1-16
more
command 1-17
Moving files 1-14
mv (move)
command 1-14, 1-15
MySQL Configuration 2-103

N

Named Startup Script 2-9
named.boot file 2-14
NFS 2-49
NFS Server Run Check 2-55
NFS-Server Structure 2-49
NIS 2-59
NIS Domain Settings 2-61
normal file (zone) 2-16

O

Online manuals (man)
command 1-35
OpenLDAP 2-44

P

Packages (rpm, turbopkg)
about 1-33
packet
about 2-25
Partitions 1-28
Password
changing 1-5
error messages 1-6
Portmap 2-91
Portmap Package Contents 2-91
PostgreSQL 2-94
PostgreSQL Configuration Parameters 2-97
Powering down 1-9
Print Spooling 2-79
Printcap 2-80
Processes (ps)
command 1-25
ProFTP 2-31
pwd (print working directory)
command 1-12

R

resolv.conf file 2-15
Resolver configuration files 2-22
Resolver Program 2-22
rm and rmdir (remove files)
command 1-16

root account 1-4
Root and swap partitions 1-28
rpm (package manager)
  using 1-33
RPM operations 2-108

S

Searching for strings (grep)
  command 1-21
server
  functions and roles 2-108
Shutting down 1-8
srm.conf file 2-39
Stopping processes (kill)
  command 1-26
Support policy xi
Switching user accounts 1-6

T

tar (creating and extracting files)
  command 1-23
TCP_Wrapper 2-108
  about 2-108
TFTP 2-29
turbopkg (package manager)
  using 1-33

U

uninstall
  RPM command 2-3
User accounts
  managing 1-4

V

Viewing text files 1-17